

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA
OSTRAVA EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Návrh a implementace e-shopu s módními oděvy
Design and Development of an E-shop with Clothes

Student:
Vedoucí bakalářské práce:

Helena Urbášková
doc. RNDr. Ivo Martiník, Ph.D.

Ostrava 2019

Zadání bakalářské práce

Student: **Helena Urbášková**

Studijní program: B6209 Systémové inženýrství a informatika

Studijní obor: 6209R017 Informatika v ekonomice

Téma: **Návrh a implementace e-shopu s módními oděvy**
Design and Development of an E-shop with Clothes

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Úvod
 2. Teoretická východiska návrhu a vývoje e-shopu
 3. Analýza současného stavu
 4. Návrh a implementace e-shopu
 5. Závěr
- Seznam použité literatury
Seznam zkratk
Prohlášení o využití výsledků bakalářské práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

GILMORE, W. J. *Beginning PHP and MySQL: From Novice to Professional*. 4th Edition. New York: Apress, 2010. ISBN 978-1-4302-3114-1.

NIXON, Robin. *Learning PHP, MySQL & JavaScript: with jQuery, CSS & HTML5*. 4th Edition. Sebastopol, CA: O'Reilly Media, 2014. ISBN 978-1491918661.

GROFF, J. R., P. N. WEINBERG and A. J. OPPEL. *SQL, the complete reference*. 3rd Edition. New York: McGraw-Hill, 2010. ISBN 0071592555.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Ivo Martiník, Ph.D.**

Datum zadání: 23.11.2018

Datum odevzdání: 10.05.2019



Ing. Petr Rozehnal, Ph.D.
vedoucí katedry



prof. Dr. Ing. Zdeněk Zmeškal
děkan fakulty

Poděkování

~~Tímto bych chtěla poděkovat vedoucímu mé bakalářské práce doc. RNDr. Ivo Martiníkovi, Ph.D. za pomoc a cenné rady.~~
Průmysly, že jsem celou práci vedle všech přírod v práci sama ostatně.

V Ostravě dne 10. května 2019


.....
Helena Urbášková

Obsah

1. Úvod.....	4
2. Teoretická východiska návrhu a vývoje e-shopu	6
2.1. Historie Internetu	6
2.2. HTML.....	7
2.2.1. HTML5.....	8
2.3. CSS.....	8
2.4. Bootstrap	9
2.5. PHP	9
2.6. JavaScript.....	10
2.7. jQuery	10
2.8. MySQL + phpMyAdmin	11
2.8.1. MySQL a SQL.....	11
2.8.2. phpMyAdmin.....	12
2.9. XAMPP	12
2.10. AJAX	12
2.11. JSON	13
2.12. UML.....	14
3. Analýza současného stavu	14
3.1. Stručný popis společnosti	14
3.2. Analýza požadavků.....	15
3.2.1. Sběr požadavků	15
3.2.2. Rozdělení požadavků	16
4. Návrh a implementace e-shopu.....	20
4.1. Návrh	20
4.1.1. Metodika Unified Process	20
4.1.2. UML diagramy	26
4.1.3. Databázový systém	36
4.1.4. Návrh vzhledu.....	39
4.1.5. Použitý software.....	42
4.2. Implementace.....	43
4.2.2.2. Provedení objednávky.....	53
5. Závěr	58
Seznam použité literatury	60

Knižní zdroje	60
Internetové + Ostatní zdroje.....	60
Seznam příloh	63

1. Úvod

V současné době je forma podnikání na živnost čím dál častějším úkazem a stále větší procento populace preferuje budování vlastní klientely a také možnost vlastní volby týkající se množství a obsahu zakázek před klasickým zaměstnáním ve státním nebo soukromém sektoru. Projekty mohou být menšího charakteru, jako například zhotovení nebo dokonce také následná implementace požadované komponenty obsažené v rámci rozsáhlejšího systému. Jak z pohledu klienta, tak i z pohledu podnikatele, je ve většině případů efektivnější varianta soustředění svých schopností na jednu danou aktivitu či oblast trhu (např. zpracování grafické části webu, copywriting, optimalizace SEO aktivit, atd.). Ve spojení s rozvojem podnikání se čím dál rychleji rozvíjí také technologické možnosti a s tím samozřejmě také digitalizace veškerých dat. Tento proces značně ovlivňuje snahu podnikatelských subjektů držet krok s konkurencí a nezaostávat ve světě moderních technologií, čímž se alespoň poměrná část přesouvá více do online prostředí. Proto je v dnešní době standardem webová prezentace, ať už v podobě statických webových stránek, či prostřednictvím dynamického řešení, jako je například internetový obchod napojený na databázi. Další neopomenutelnou novinkou v minulých letech je nutnost přizpůsobení webových stránek pro různé typy zařízení a také díky odlišnému přístupu firmy Apple a jeho operačního systému iOS pro různé typy webových prohlížečů. Důležitým a často zásadním faktorem při udržení zákazníka na dané webové stránce je její doba odezvy a jednoduchost doprovázená intuitivností manipulace a orientace na stránce.

Transformace našeho podnikání do podoby dynamického řešení prostřednictvím spojení databáze, komunikace s webovým serverem, tvorba GUI v podobě webových stránek a možnosti okamžitého nákupu je v dnešní době stále častěji využívána. Po několik schůzkách se slečnou Rabiečnou byly stanoveny přesně dané požadavky na výslednou podobu požadovaného produktu, na jejímž základě byly stanoveny následující cíle, které byly rozděleny do tří základních kategorií:

Analýza

- sběr poznatků a informací nezbytných pro následné definování požadavků zadavatelky,
- provedení analýzy těchto požadavků a jejich následné rozdělení na požadavky funkční a nefunkční,

- analýza konkurenčního prostředí.

Návrh

- výběr vhodné metodiky pro navrhnutí a vývoj internetového obchodu,
- specifikace a modelování případu užití s využitím UML diagramů,
- návrh databázové struktury,
- návrh uživatelského rozhraní,
- specifikace aplikovaného softwaru.

Implementace

- popis uživatelského rozhraní,
- ukázka implementace vybraných funkcionalit.

2. Teoretická východiska návrhu a vývoje e-shopu

2.1. Historie Internetu

"Počátky Internetu sahají do roku 1960, kdy se rozběhl projekt Ministerstva obrany USA pod názvem Advanced Research Project Agency (ARPA). Jedním z cílů tohoto projektu bylo propojit počítače pomocí sítě, a to rychle a na velké vzdálenosti. Tato síť byla navržena jako zabezpečená síť, díky čemuž spolu mohly počítače komunikovat i v případě, kdy se různé trasy mezi nimi staly neprůchodné. V roce 1969 se pak zrodil tzv. ARPANet, který tímto způsobem propojoval několik klíčových univerzit."[1]

Na začátku 70. let byl vytvořen protokol TCP/IP (Transmission Control Protocol/Internet Protocol), který se rychle rozšířil jako standard pro výměnu paketů v síti ARPANet. Přibližně ve stejné době se začínal tento protokol uplatňovat také k propojování menších počítačových sítí za účelem možnosti sdílení dat na národní, či dokonce celosvětové úrovni.

Na počátku 90. let začal být Internet využíván i velkými společnostmi pro emailovou komunikaci, nicméně jeho komerční nasazení vzápětí zakázala Národní vědecká nadace USA. Tento zákaz byl však posléze zrušen, a tak se postupem času Internet stal vzdělávacím prostředkem i komerčním médiem pro přímý prodej a reklamu [2]. Koncem 90. let dochází k zániku ARPANETu a na jeho místo nastupuje síť NSFNET (National Science Foundation NETwork). Roku 1989 se objevuje první návrh standardu WWW (World Wide Web). W3 je označení pro aplikace internetového protokolu HTTP, v dnešní době již spíše HTTPS (spojení protokolů HTTP a SSL/TLS pro šifrovanou komunikaci), které tvoří soustavu propojených hypertextových dokumentů. Autorem tohoto standardu je konsorcium W3C (World Wide Web Consortium), které se věnuje dalšímu vývoji webových technologií.

13. února 1992 se poprvé k Internetu připojila také Česká republika, a to konkrétně budova ČVUT v Praze. Na přelomu roku 1992/1993 vzniká celorepubliková páteřní síť CESNET. Následně roku 1993 začíná vývoj prvního internetového prohlížeče Mosaic společnosti Mosaic Communications, nyní Netscape Communications.

V následujících letech nastal pro Internet bouřlivý vývoj, který pokračuje dodnes. Každoročně se zvyšuje počet uživatelů připojených ke světové síti a zároveň rostou možnosti využití, kde svou roli hrají různé druhy komunikace ať už v podobě e-mailů či telefonování přes internet. [9]

2.2. HTML

Jazyk HTML je formálně definován jako reprezentační jazyk popisující především typografickou a logickou strukturu dokumentů systému SGML (Standard Generalized Markup Language). Zkratka HTML reprezentuje anglický název HyperText Markup Language, což znamená, že webové stránky tvořené v tomto jazyce jsou mezi sebou provázány hypertextovými odkazy.

První definice HTML se objevila roku 1991 jako součást projektu WWW, který byl vyvíjen britským fyzikem a počítačovým inženýrem Timem Bernesem-Leem. Celý projekt vznikl ve švýcarském CERNu (Evropské centrum jaderného výzkumu). Hlavním účelem tohoto projektu bylo vyřešení problému se sdílením informací a dat ve velkých institucích.

HTML jazyk se začal v následujících letech postupně vyvíjet. Další podstatnou verzí byla HTML 3.2, která byla představena roku 1996 a obohacovala stávající možnosti jazyka o tabulky, zarovnání textu a stylové prvky při ovlivňování vzhledu. Roku 1999 nastupuje sofistikovanější verze HTML 4 se svými novými prvky tabulek, formulářů a přidání rámců. Původně se počínaje verzí 4 předpokládalo zastavení vývoje jazyka HTML. Z tohoto důvodu se z této verze HTML se přešlo na standard XHTML, což je ve své podstatě HTML4 se striktní syntaxí ve stylu XML. Nicméně tento krok se ukázal jako chybný, a tím se dostáváme k datu 28. října 2014, kdy nastupuje nejnovější verze jazyka HTML5 a také její XML varianta XHTML5.

2.2.1. HTML5

Tato verze jazyka HTML zahrnuje všechny platné prvky jak z HTML4, tak i z XHTML 1.0. Tento standard byl navržen tak, aby bylo zajištěno, že bude fungovat téměř na jakékoliv platformě. [3]

Od té doby jeho oblíbenost neustále roste, a to zejména díky novým možnostem, které nám přináší. Vývojáři hojně používají HTML5 zejména kvůli jeho novým nástrojům, díky kterým mohou vytvářet lepší uživatelská rozhraní. Mezi nejznámější rozšíření patří např. přímá podpora multimédií v prohlížeči, podpora aplikací bez připojení k Internetu, popisnější značky, lepší komunikace mezi weby či okny, animace, stíny, různé přechody, transformace a mnohé další velice zajímavé funkčnosti. [10]

Jedna z mnoha změn se týká specifikace typu dokumentu, tedy DOCTYPE HTML. Tato deklarace říká prohlížečům, jaký typ dokumentů se chystá zpracovávat. Původně byl DOCTYPE velice zdlouhavý a špatně zapamatovatelný. Oproti předchozí verzi HTML4 z roku 1997 se jeho zápis výrazně zkrátil a došlo k jeho zjednodušení jak v psané formě, tak i z hlediska zapamatovatelnosti. Mezi základní cíle deklarace DOCTYPE patří především snaha o usnadnění práce validátorům ve smyslu určení, jaký typ validace je zapotřebí zvolit při ověřování kódu. [4]

DOCTYPE

- HTML4

```
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
  Transitional//EN" "  
  http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.">
```

- HTML5

```
- <!DOCTYPE html>
```

Obrázek 2.1: Wireframe – Úvodní stránka [www.slideshare.net]

2.3. CSS

Jazyk CSS (Cascading Style Sheets) je kolekce stylů pro úpravu vzhledu webových stránek pomocí popisu způsobu, jakým se dané elementy na webové stránce napsané v jazycích HTML, XHTML nebo XML mají zobrazovat. Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Hlavní cílem kaskádových stylů je oddělit vzhled dokumentu od jeho struktury a obsahu. Separace logické struktury a vzhledu dokumentu vystřídala HTML značky stylu, jako např. <big>, nebo <center>, které byly obsaženy právě přímo v HTML dokumentu. Tento způsob stylování byla značná nevýhoda v případě,

jestliže jste chtěli použít jeden styl vícekrát, protože nebyla jiná možnost, než tento styl napsat na každou stránku dokumentu zvlášť. Díky připojení jednotného CSS dokumentu ke všem stránkám dokumentu pomocí odkazu stačí nyní přepsat pouze jeden výskyt stylu v našem CSS dokumentu a změny se projeví všude, kde se daný styl nachází.

Nejnovější verze tohoto jazyka je CSS 3. Jedná se o vylepšení předchozí verze (CSS 2.1), do které bylo přidáno mnoho nových a zajímavých funkcionalit. [3] Účelem těchto funkcionalit je usnadnění a zpřehlednění práce webových designérů. Mezi nejznámější novinky řadíme např. podporu dodatečných selektorů, vržené stíny, přechody, vícenásobná pozadí a také různé druhy animací.

2.4. Bootstrap

Bootstrap představuje nejoblíbenější bezplatný framework pro HTML, CSS a JavaScript, který umožňuje vývoj responzivních webových stránek optimalizovaných pro mobilní zařízení. Framework zahrnuje responzivní šablony CSS a HTML pro tlačítka, tabulky, navigaci, otočné obrázky a další elementy použitelné na webové stránce. K dispozici je vám i několik volitelných zásuvných modulů JavaScript, se kterými mohou i vývojáři s pouze základní znalostí psaní kódu vytvářet špičkové responzivní webové stránky. [11]

Bootstrap, původně pojmenovaný Blueprint, vytvořil Mark Otto a Jacob Thornton na Twitteru jako framework podporující konzistenci mezi interními nástroji. Před Bootstrapem byla pro vývoj rozhraní využívána řada knihoven, což vedlo k nekonzistenci zdrojových kódů a velké náročnosti na údržbu. V srpnu 2011 byl vydán Twitter Bootstrap jako open source a v únoru 2012 se stal nejoblíbenějším developerským projektem na GitHubu. Výhodou tohoto souboru nástrojů je snadné zpracování jakéhokoliv uživatelského rozhraní ve webové aplikaci a nerozhoduje, zda to je například uživatelské rozhraní v administraci back-endových nebo front-endových aplikací. Bootstrap je kompatibilní s poslední verzí všech hlavních prohlížečů a elegantně se přizpůsobuje použití na starších prohlížečích jako je Internet Explorer 8. Od verze 2.0 také podporuje responzivní design. [12]

2.5. PHP

PHP (Hypertext Preprocessor) je skriptovací multiplatformní programovací jazyk, který pracuje na straně webového serveru Apache. Hojně se využívá především při tvorbě dynamických webových stránek (obsah stránky je dynamicky generován na základě interakce uživatele) a také webových aplikací užívajících HTML, XHTML či WML. Při odeslání požadavku klienta na server se na jeho straně spustí skript, který nám často ve spolupráci s daným

databázovým systémem (nejčastěji MySQL) zašle odpověď ve formátu dokumentu HTML s požadovaným dynamicky vygenerovaným obsahem.

Autory PHP jsou dva izraelští vývojáři Zeev Suraski a Andi Gutmans, kteří vytvořili základ PHP v roce 1997. Jeho nejnovější verze, tedy verze 7.3, byla spuštěna 6. prosince 2018. Přináší různá vylepšení, jakými jsou např. větší flexibilita syntaxe Heredoc (umožňuje přidat velké množství textu bez nutnosti užití dvojitého uvozovky), Nowdoc či implementaci velice potřebné `JSON_THROW_ON_ERROR` metody pro obstarávání JSON chyb.

2.6. JavaScript

JavaScript je multiplatformní skriptovací objektově orientovaný programovací jazyk, který vznikl roku 1995 a jeho autorem je Brendan Eich. Jedná se o programovací jazyk, který je vykonáván na straně webového klienta i serveru a primárně slouží především ke tvorbě dynamických webových stránek. Pomáhá nám vytvořit různé vizuální efekty, které dokáží oživit webové stránky, případně definovat funkce, které se uskuteční při určitých reakcích uživatele. (Odell, 2010) JavaScript je možno využít i na straně serveru, např. v podobě open source softwarových řešení, jejichž typickým zástupcem je např. Node.js.

K rychlejší a efektivnější implementaci jazyka JavaScript se také často používají různé APIs. Zkratka APIs je označení pro Application Programming Interfaces, které nám umožňují použít hotové kolekce stavebních kódů, pomocí kterých lze implementovat funkce, které bychom bez využití patřičného API nebyli schopni implementovat. Jedním z velice podstatných API webového prohlížeče v případě JavaScriptu je DOM (Document Object Model) API umožňující manipulaci s HTML či XML dokumenty zahrnujících přístup či modifikaci obsahu, struktury, stylu dokumentu nebo jeho části. V minulosti měl každý webový prohlížeč unikátní specifické rozhraní určené k manipulaci s HTML elementy. Vzájemná nekompatibilita těchto rozhraní byla důvodem pro jejich standardizaci. Této úlohy se ujalo konsorcium W3C, které vytvořilo W3C Document Object Model, který je nezávislý jak na platformě, tak i na programovacím jazyce.

2.7. jQuery

jQuery je jedna z nejoblíbenějších javascriptových knihoven, která klade důraz na jednoduchost, čitelnost a rychlost kódu. Funkce knihovny je možno implementovat multiplatformně a zcela zdarma. Vytvoření jQuery se datuje k 20. lednu 2006, přičemž autorem je softwarový inženýr John Resig. Na knihovně jQuery je založena také knihovna jQuery UI, kterou v roce 2007 naprogramoval Paul Bakalus. Tato knihovna nám usnadňuje tvorbu různých dynamických prvků uživatelského rozhraní. [10]

jQuery je postavena na filozofii, která říká, že stejně jako CSS odděluje stylové prvky od HTML kódu, tak jQuery separuje chování příslušného elementu od struktury HTML dokumentu. K funkcím knihovny jQuery se přistupuje s prostřednictvím funkce jQuery, jejíž zkratka, která je daleko více používaná, je definována jako `$ ()`.

2.8. MySQL + phpMyAdmin

Žádný větší webový projekt by se neobešel bez patřičného databázového systému, který zabezpečuje ukládání článků, produktů, uživatelských informací a mnoha dalších dat, které se obvykle ukládají ve formě tabulek.

2.8.1. MySQL a SQL

SQL

SQL neboli Structured Query Language reprezentuje strukturovaný dotazovací jazyk, který se užívá pro práci s daty v relačních databázích. Cílem jazyka SQL je aplikace příkazů, které se svou syntaxí co nejvíc blíží přirozenému jazyku, a také proto je databázový jazyk SQL velice intuitivní a jednoduchý na pochopení. Prostřednictvím jazyka SQL můžeme provádět příkazy, které umožňují úplnou kontrolu nad SŘBD a jsou rozděleny do čtyř základních skupin:

- příkazy pro manipulaci s daty (SELECT, INSERT, UPDATE, DELETE, ...),
- příkazy pro definování dat (CREATE, ALTER, DROP, ...),
- příkazy pro řízení přístupových práv (GRANT, REVOKE, ...),
- příkazy pro řízení transakcí (START TRANSACTION, COMMIT, ROLLBACK, ...).

Pátou skupinu tvoří tzv. ostatní příkazy, pomocí nichž lze do databáze přidávat uživatele nebo také nastavovat její systémové parametry.

MySQL

MySQL je populární multiplatformní databáze, jejíž komunikace probíhá prostřednictvím dotazovacího jazyka SQL. Hlavními autory MySQL jsou Michael Widenius a David Axmark. MySQL je distribuována ve formě bezplatné GPL licence, ale také v komerční placené verzi. MySQL databázový systém je velmi populární v bezplatné verzi na serverech Apache běžících pod operačním systémem Linux v kombinaci s PHP. Tato velmi výkonná, stabilní a zároveň bezplatná technologická kombinace je známá pod zkratkou LAMP. [16]

Nejnovější verze MySQL je verze 8.0 vydaná 19. dubna 2018, ve které došlo ke značnému zrychlení výkonu databáze jak při zápisu, tak i při čtení, a také mimo jiné ke zlepšení bezpečnosti výchozí autentizace.

2.8.2. phpMyAdmin

Aplikace phpMyAdmin je multiplatformní nástroj napsaný v jazyce PHP, který umožňuje jednoduchou správu obsahu databáze prostřednictvím webového rozhraní. Současná verze 4.8.5 vydána v letošním roce umožňuje vytvářet/rušit databáze, vytvářet/modifikovat/rušit tabulky a také provádět SQL příkazy ve spolupráci se správou klíčů. [13]

Zakladatelem je Tobias Ratschiller, který v PHP začal psát webový frontend k MySQL inspirovaný MySQL-Webadminem od Petere Kuppelwiesera. Když projekt v roce 2000 opustil kvůli nedostatku času, phpMyAdmin se již stal jednou z nejpoužívanějších aplikací v PHP a nástrojem pro správu MySQL s velkou komunitou uživatelů a vývojářů. [13] Pro pokračování vývoje zaregistrovala v roce 2001 skupina tří vývojářů – Olivier Müller, Marc Delisle a Loïc Chapeaux – projekt phpMyAdmin na SourceForge a začala vydávat nové verze.

2.9. XAMPP

Jedná se o multiplatformní softwarový balíček vyvinutý firmou Apache Friends, jehož obsahem je volně dostupný open-source software (webový server Apache, databáze MariaDB a interpret programovacích jazyků PHP a Perl). Každé písmeno z názvu XAMPP reprezentuje jednu výše zmíněnou funkci: X = Multiplatformní, A = Apache, M = MariaDB, P = Perl. XAMPP je jednoduchá odlehčená distribuce Apache, která pro vývojáře velmi jednoduše vytvoří lokální webový server pro vývoj a testování. Vzhledem k tomu, že většina implementací webových serverů využívá stejné komponenty jako XAMPP, je přechod z lokálního testovacího serveru na produkční server velmi snadný. [14]

2.10. AJAX

AJAX neboli zkratka pro Asynchronous JavaScript and XML zastřešuje soubor technologií kombinující JavaScript, HTML DOM (Objektový model + API pomocí něhož a následně za užití JavaScriptu lze modifikovat HTML elementy) a XMLHttpRequest object, který je implementován v každém prohlížeči (pro vyžádání dat z webového serveru).

Největší kouzlo AJAXu spočívá v tom, že nám umožňuje aktualizaci webové stránky asynchronně prostřednictvím výměny dat mezi klientem a webovým serverem tzv. “za oponou“. To znamená, že je možná aktualizace části nebo celé webové stránky bez nutnosti znovunačtení celé webové stránky, což výrazně zvyšuje rychlost práce s touto stránkou.

Jak AJAX vlastně funguje si nyní zjednodušeně popíšeme v následujících sedmi krocích:

1. Nastane spouštěcí událost – načtení stránky, kliknutí na tlačítko/odkaz,
2. Je zabezpečeno vytvoření XMLHttpRequest objektu prostřednictvím JavaScriptu,
3. XMLHttpRequest objekt zašle požadavek na webový server,
4. Webový server požadavek zpracuje,
5. Strana klienta obdrží odezvu/reakci od serveru,
6. Odpověď je zpracována JavaScriptem,
7. Následuje provedení požadavku strany klienta (aktualizace stránky) prostřednictvím JavaScriptu.

2.11. JSON

JSON (JavaScript Object Notation) je odlehčený formát pro výměnu dat. Je jednoduše čitelný i zapisovatelný uživatelem a snadno analyzovatelný i generovatelný programově. JSON je založen na vybrané podmnožině programovacího jazyka JavaScript, Standard ECMA-262 3rd Edition December 1999. JSON je textový na programovacím jazyce zcela nezávislý formát využívající konvence dobře známé programátorům jazyků rodiny C (C, C++, C#, Java, JavaScript, Perl, Python a dalších). Díky tomu je JSON pro výměnu dat opravdu ideálním formátem.

JSON je založen na dvou strukturách:

- kolekce párů název/hodnota, která bývá v rozličných programovacích jazycích realizována jako objekt, záznam (record), struktura (struct), slovník (dictionary), hash tabulka, klíčový seznam (keyed list) nebo asociativní pole,
- seřazený seznam hodnot, který je ve většině jazyků realizován jako pole, vektor, seznam (list) nebo posloupnost (sequence).

Jedná se o univerzální datové struktury a v podstatě všechny moderní programovací jazyky je v nějaké formě podporují. Je tedy logické, aby na nich byl založen i na jazyce nezávislý výměnný formát.

V JSON jsou tyto struktury realizovány s využitím následujících konstrukcí:

Objekt je neuspořádaná množina párů název/hodnota. Objekt je uvozen znakem { (levá složená závorka) a zakončen znakem } (pravá složená závorka). Každý název je následován znakem : (dvojtečka) a páry název/hodnota jsou pak odděleny znakem , (čárka). [15]

2.12.UML

Jazyk UML (Unified Modeling Language) je univerzální jazyk pro vizuální modelování systémů, který se používá ke specifikaci, vizualizaci, konstrukci a dokumentaci artefaktů softwarového systému. Používá se k pochopení, návrhu, prohlížení, konfiguraci, údržbě a kontrole informací o systémech. Byl vyvinut se záměrem možnosti použití se všemi vývojovými metodami, fázemi životního cyklu, aplikačními doménami a médii. [6] Je nesmírně důležité si uvědomit, že jazyk UML nenabízí žádný druh metodiky modelování. Samotný jazyk UML poskytuje pouze vizuální syntaxi, kterou můžeme aplikovat při vývoji svých modelů. [7] Je určen k podpoře při iterativním procesu vývoje, ale také k většině existujících procesů objektově orientované vývoje. UML zachycuje informace o statické struktuře a dynamickém chování systémů. Systém je modelován jako soubor diskrétních objektů, které interagují za účelem vykonávání akcí. Statická struktura definuje druhy objektů, které jsou důležité pro systém a jeho implementaci, jakož i vztahy mezi objekty. Dynamické chování definuje historii objektů v čase a komunikaci mezi objekty k dosažení cílů. Obsahuje konstruktory pro reprezentaci implementačních rozhodnutí a pro organizaci runtime prvků do komponent. UML není určen k modelování spojitých systémů nacházejících se ve strojírenství a fyzice. UML má být univerzální pro všeobecné použití jako modelovací jazyk pro diskrétní systémy, kterými jsou např. softwarové systémy. Struktura jazyka UML se skládá ze tří základních částí: stavební bloky, společné mechanismy a architektura. Stavební bloky poté tvoří předměty (things), vztahy (relationships) a digramy (diagrams). Relace umožňuje ukázat na modelu, jaký je vztah mezi dvěma předměty. Příkladem relace může být závislost (dependency), asociace (association), agregace (aggregation), kompozice (composition) a další. [6]

3. Analýza současného stavu

3.1. Stručný popis společnosti

Projekt je zhotoven na zakázku pro slečnu Bc. Simonu Rabiečnou, která pracuje na HPP na pozici grafického designéra ve společnosti PEMA CAR s.r.o. a zároveň má také status OSVČ. Byl zhotoven podnikatelský záměr na téma založení a provozování internetového obchodu s módními doplňky, nejpravděpodobněji v rámci značky Vans. Současná verze internetového obchodu je nyní již schválena zadavatelkou a implementována v praxi.

3.2. Analýza požadavků

Klíčovou roli v každém vývojovém procesu jakéhokoliv produktu hraje fáze určení potřeb uživatele. Proces vývoje specifikace potřeb se nazývá Requirements Engineering, zahrnující tyto aktivity:

- zkoumání činnosti – aktivit,
- snaha o porozumění uživatelských potřeb s cílem vylepšení programového vybavení,
- transformace potřeb do formy přesných a jasně definovaných požadavků.

3.2.1. Sběr požadavků

Požadavky na budoucí podobu internetového e-shopu byly konzultovány se slečnou Bc. S. Rabiečnou a po několika společných konzultacích a prodiskutování veškerých požadavků na systém jsme si požadavky na systém specifikovaly v podobě následujících kritérií:

- jednoduché vkládání, správa a možnost odebrání nabízeného zboží,
- možnost editace informací o zboží, správa uživatelů, správa kategorií zboží,
- separování přístupových práv person v podobě administrátora, editora,
- omezené možnosti pro editora,
- přítomnost nákupního košíku,
- snadná úprava designu stránek,
- ošetření nestandardních situací.

Mimo výše jmenované požadavky je také důležité nezapomenout na obecné požadavky, které by měly být aplikovány na každou webovou prezentaci:

- přehlednost a intuitivní ovládání e-shopu,
- přístupnost pro širokou škálu návštěvníků,
- responsivní design.

3.2.2. Rozdělení požadavků

V rámci specifikace požadavků je nutné získané uživatelské požadavky rozdělit do následujících tří základních skupin:

3.2.2.1. Funkční požadavky

Funkční požadavky specifikují, co má daný programový systém dělat, tj. služby a funkcionality, které by měl systém nabízet, a způsob, jakým by měl systém spolupracovat se svými uživateli. Funkční požadavky na náš systém jsou pak následující:

- jednoduché vkládání, správa a možnost odebrání nabízeného zboží,
- možnost editace informací o zboží, správa uživatelů, správa kategorií zboží,
- separování přístupových práv person v podobě administrátora, editora,
- omezené možnosti pro editora,
- přítomnost nákupního košíku,
- možnost platby kartou pomocí platební brány GoPay,
- seznam objednávek,
- snadná úprava designu stránek,
- ošetření nestandardních situací – vyhození výjimek.

3.2.2.2. Nefunkční požadavky

Specifikují omezení, jak má systém fungovat, a standardy pro jeho provoz. Většinou souvisí s implementačním prostředím podniku, tj. konkrétním typem hardwaru nebo softwaru (např. typ databáze, druh použitého hardwaru). V našem případě se jedná o následující nefunkční požadavky:

- přehlednost a intuitivní ovládání e-shopu,
- jednoduchost jeho použití,
- stabilita internetového obchodu,
- přístupnost pro širokou škálu návštěvníků,
- responsivní design,

- rychlost načítání stránky – max. hodnota indexu rychlosti 3 s.,
- šifrovaná forma hesel v databázi.

3.2.2.3. Analýza konkurence

Často velmi podceňovanou složkou je analýza konkurence. Rozsah konkurenčního trhu bychom mohli rozdělit do dvou základních segmentů:

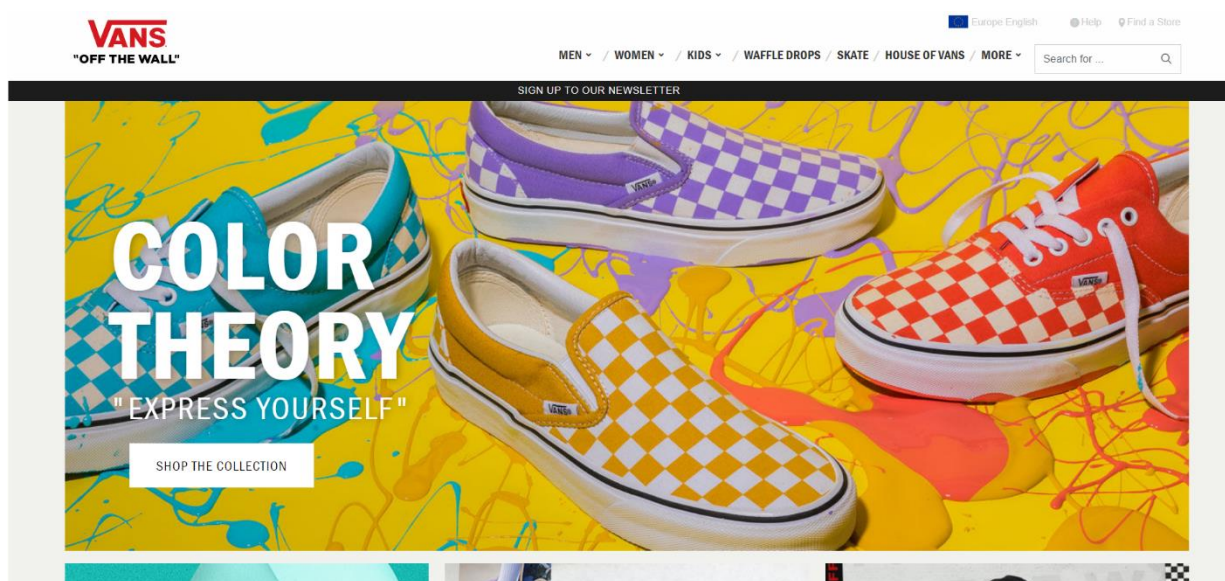
1. velmi velký počet konkurentů na trhu + středně nasycený trh – o to větší nutnost odlišení naší firmy od ostatních konkurentů,
2. bezkonkurenční trh – v tomto případě nás mohou čekat vysoké náklady na marketing, neboť není-li námi nabízené zboží známé na trhu, nastává nutnost nalezení té správné cesty k zákazníkovi a nutnost schopnost přesvědčení jej o tom, že právě náš produkt stojí za to si koupit.

V rámci analýzy konkurenčních firem jsou doporučovány tyto kroky:

- přihlášení se k odběru konkurenčního newsletteru,
- sledování sociálních sítí,
- sledování změn,
- cenová politika, zavádění novinek na trh a prezentace slev, či akcí,
- doplňkové služby (tyto vedlejší služby jsou často důvodem k opětovnému nákupu a spokojenosti zákazníka),
- funkce konkurenčních e-shopů,
- objednání zboží + sledování procesu zpracování objednávky,
- reklamní kampaně,
- recenze zákazníků,
- nalezení konkurenčních výhod, kterými se prezentují. [8]

VANS Official site

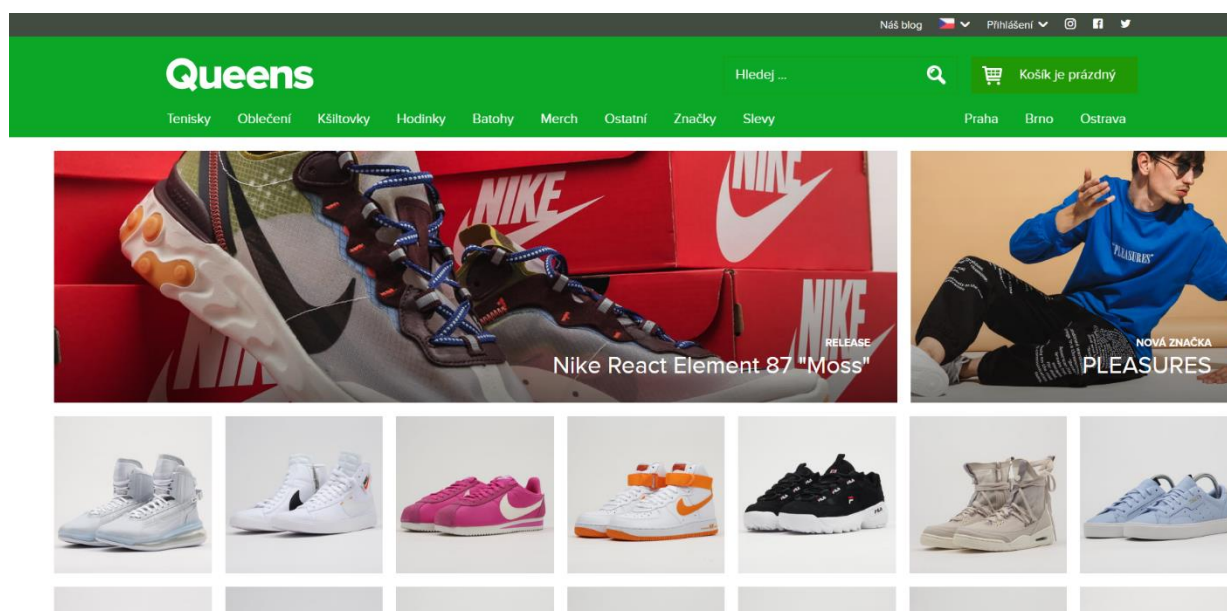
Vans je americká firma známá hlavně svou obuví, která byla založena roku 1966 se sídlem v Kalifornii. Přestože je tato firma velice známá pro svou obuv, velice oblíbené se těší také její oblečení a doplňky. Jeden z typických prvků specifických pouze pro tuto značku je možnost návrhu designu vlastní obuvi. Skutečný zájem o zákaznickovy potřeby je u firmy Vans patrný i po provedení nákupu v podobě nákladů na dopravu zdarma a také v případě vrácení zboží není potřeba za nic platit. Firma funguje také prostřednictvím kamenných prodejen, dostupných po celé Evropě a své produkty prodává také v pobočkách obchodů jako jsou např. D-Sport, Meatfly, apod.



Obrázek 3.1: Vans e-shop – Úvodní stránka [www.vans.eu]

Queens

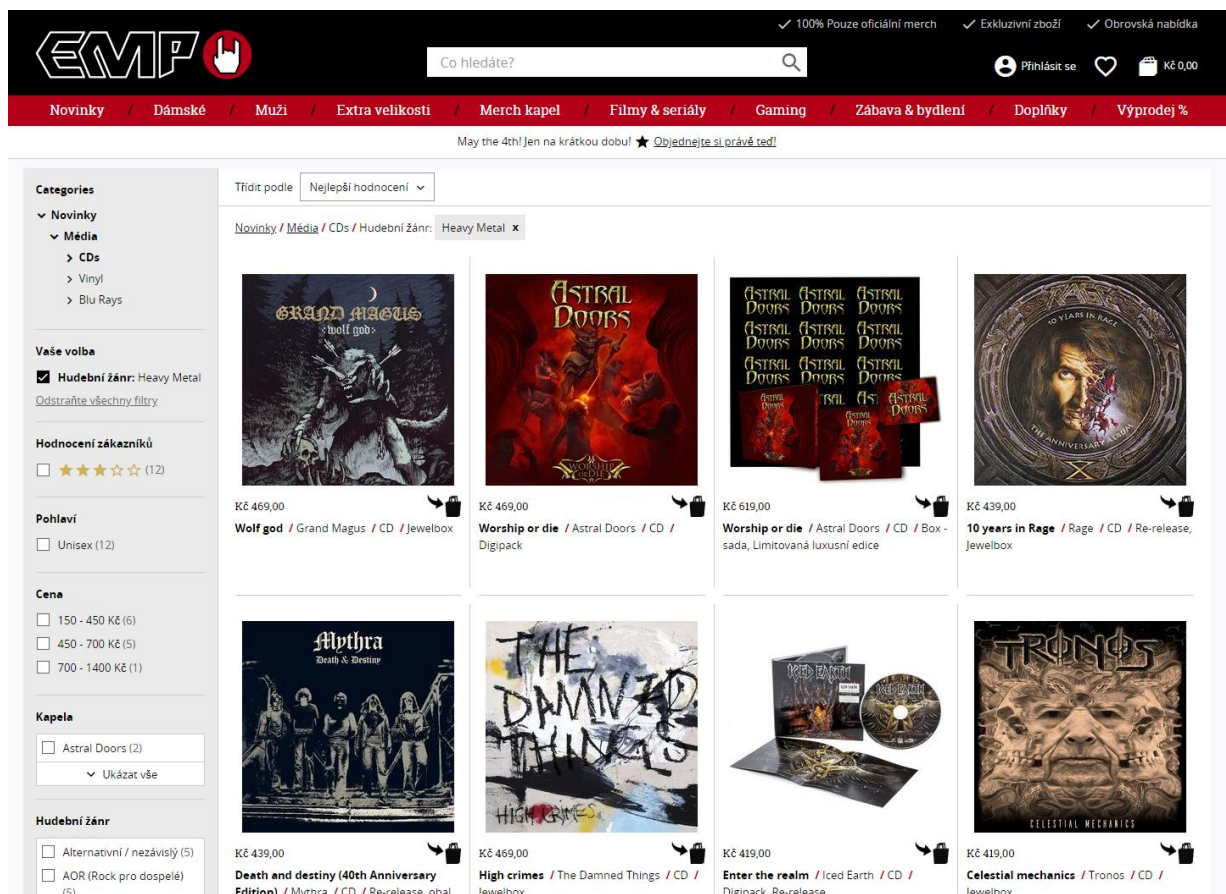
Queens je mezinárodní řetězec módních obchodů a e-shop zaměřující se výhradně na prodej streetwear módy a tenisek. Firma vznikla relativně nedávno, v roce 2003 v Ostravě, nicméně po krátké době je firma Queens jednou z nejznámějších firem ve svém odvětví v ČR. S touto firmou mám osobní zkušenost, kterou hodnotím velice kladně. Po výběru zboží vám je doručen e-mail o potvrzení objednávky a shrnutím informací o objednávce. Jakmile je objednávka připravena k odeslání či vyzvednutí, obdržíte na prodejně další e-mail s touto informací. Doba doručení je ve většině případů do tří pracovních dnů. Své pobočky má firma v Praze, Brně a Ostravě. Možnost vrácení zboží je do čtrnácti dní zdarma.



Obrázek 3.2: Queens e-shop – Úvodní stránka [www.queens.cz]

EMP Merchandising

Jedná se o německý internetový obchod založený roku 1986 a nabízející širokou škálu produktů zahrnující oblečení, obuv, ale také CD a DVD alba se zaměřením na rockový a metalový žánr. Z hlediska jazykových možností je internetový obchod firmy perfektní volbou pro zákazníky z celé Evropy díky tomu, že si stránky lze zobrazit až v patnácti jazykových verzích. Také s tímto e-shopem mám pozitivní osobní zkušenost. Proces zpracování objednávky zahrnuje okamžitě po potvrzení objednávky obdržení e-mailu se shrnutím informací o Vaší objednávce, poté obdržíte také e-mail při odeslání zboží. Doručení objednávky trvá cca. čtyři pracovní dny. Co se mně osobně nelíbí je pak skutečnost, že jestliže se jednou přihlásíte k odběru newsletteru firmy, pak dostáváte e-maily i dvakrát denně, nicméně funkce EMP Newsletter se dá vždy odhlásit. Firma funguje pouze na online prodeji, nicméně doručuje zboží do Rakouska, Finska, Španělska a mnoho dalších zemí.



Obrázek 3.3: EMP e-shop – Úvodní stránka [www.emp-shop.cz]

4. Návrh a implementace e-shopu

V následující kapitole se budeme věnovat popisu klíčových částí internetového obchodu – jejich návrhu a následné implementaci vybraných postupů. Budeme se zabývat návrhem databáze a charakteristikou aplikovaných metod realizovaných za účelem plné funkčnosti daných funkcionalit.

4.1. Návrh

4.1.1. Metodika Unified Process

4.1.1.1. Obecný popis

Metodika UP – Unified Process (unifikovaný nebo jednotný proces) vychází z průmyslového standardu SEP – Software Engineering Process. UP je jen zkrácené označení průmyslového standardu USDP – Unified Software Development Process (unifikovaný proces vývoje software). Jednoduše řečeno, jedná se o generický proces pro jazyk UML – Unified Modeling Language (unifikovaný modelovací jazyk). Jde pouze o základní, generickou a otevřenou metodiku, kterou je možné uzpůsobit jakémukoli rozsahu projektu.

Unifikovaný proces (UP) znamená:

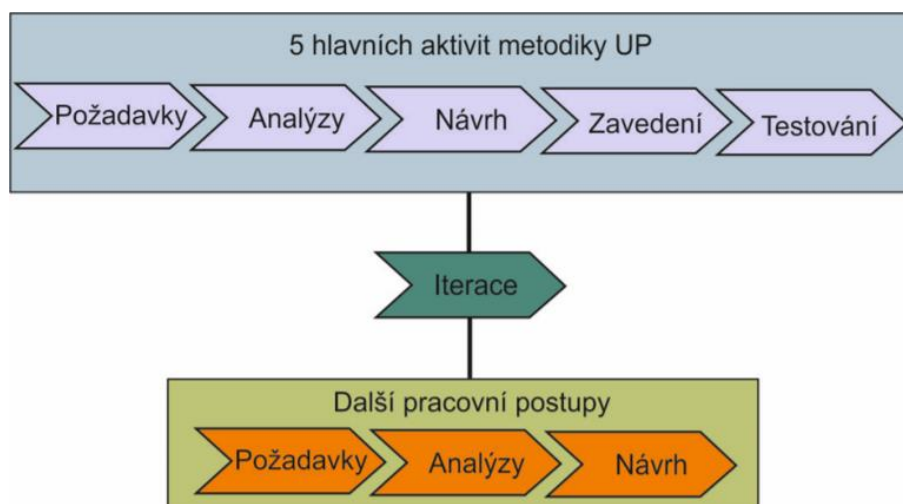
- řízení požadavky a případy užití,
- řízení rizikem,
- základ na architektuře,
- iterativní a přírůstkový proces vývoje SW produktu.

4.1.1.2. Iterativní a inkrementační přístup

UP je rozdělen do jednotlivých iterací, z nichž každá prochází pěti základními pracovními procesy:

- stanovení požadavků,
- analýza,
- návrh,
- implementace,
- testování,
- iterace v UP.

Jednotlivé iterace mohou probíhat i paralelně. To dovoluje souběžnost a flexibilitu prací u velkých projektů. [17]



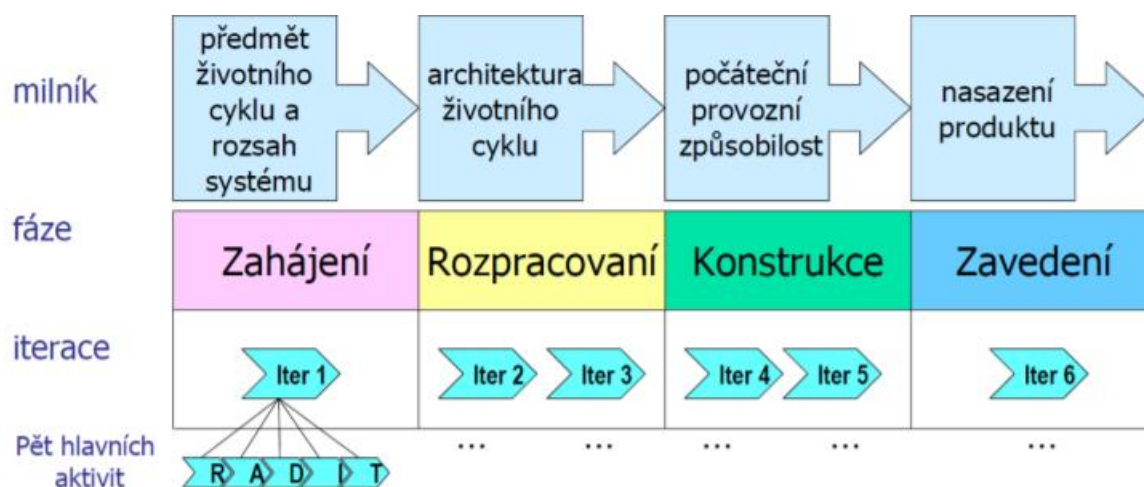
Obrázek 4.1: Metodiky vývoje softwaru – Fáze aktivit metodiky UP [www.mvso.cz]

Iterace a přírůstky

Každá iterace generuje vlastní základní linii (baseline), která se skládá z částečně kompletní verze finálního systému a z veškeré přidružené projektové dokumentace. Zmiňované základní linie jsou postupně vrstveny tak dlouho, dokud není dosažena konečná podoba vytvářeného systému.

Rozdíl mezi dvěma liniemi je označován jako přírůstek (inkrement). Z tohoto důvodu je životní cyklus projektů podle metodiky UP označován jako iterativní a přírůstkový (inkrementační).

4.1.1.3.Struktura UP



Obrázek 4.2: Metodiky vývoje softwaru – Struktura UP [www.mvso.cz]

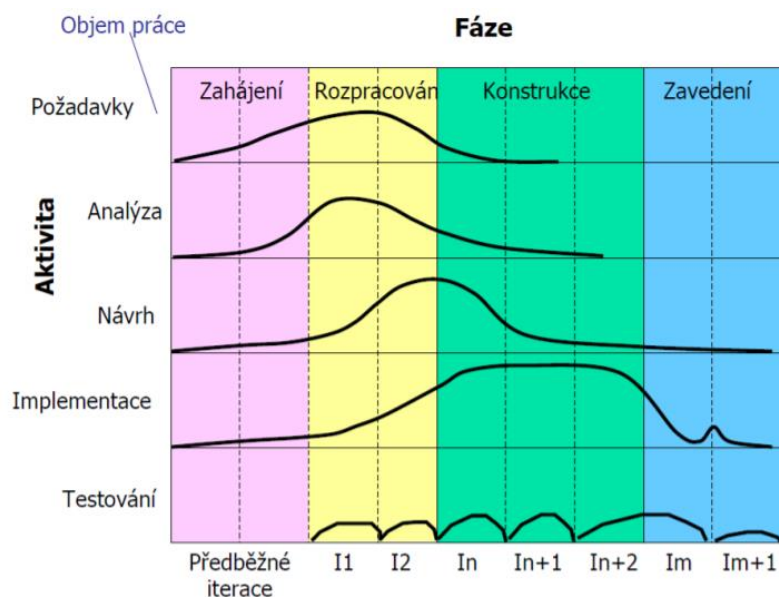
Každá etapa (fáze) může být tvořena jednou či více iteracemi. Kolik takovýchto iterací bude se vždy odvíjí od velikosti daného projektu. Každá etapa je ukončena tzv. milníkem (milestone).

U každé z etap je nezbytné zvážit:

- základ pracovního postupu,
- cíl(-e) etapy,
- konečný milník etapy.

Jak je zřejmé z obrázků 4.1 a 4.2, každá aktivita v UP má několik fází:

- zahájení,
- rozpracování,
- realizace,
- zavedení.



Obrázek 4.3: Metodiky vývoje softwaru – Fáze aktivit metodiky UP [www.mvso.cz]

Zabývejme se nyní těmito fázemi podrobněji.

Činnosti fáze zahájení:

- **požadavky** – zvážení důvodů projektu, jeho přínosů a nejdůležitějších požadavků,
- **analýza** – při této činnosti je analyzována a stanovena proveditelnost,
- **návrh** – navrhují se určité technické prototypy,
- **implementace** – určují a vytváří se koncepce technických prototypů,
- **testování** – ve fázi zahájení se testování neprovádí.

Cíle fáze zahájení:

- stanovení proveditelnosti projektu, vytvoření koncepcí a technických prototypů,
- tvorba obchodních případů,
- určení klíčových požadavků a přínosu systému,
- určení kritických rizik.

Milník fáze zahájení:

- definovány klíčové požadavky, které musí být odsouhlaseny investorem,
- definovány systémové vlastnosti,
- tvorba spustitelného architektonického základu,

- odhad rizik,
- obchodní případy,
- investor souhlasí s cílem projektu.

Činnosti fáze rozpracování:

- požadavky – upřesnění rozsahu systému a požadavků, které na něj jsou a budou kladeny,
- analýza – při této činnosti se stanovuje, co budeme vytvářet,
- návrh – vytvoření stabilní architektury,
- implementace – vytvoření spustitelného architektonického základu.
- testování – testování spustitelného architektonického základu.

Cíle fáze rozpracování:

- vytvoření spustitelného architektonického základu,
- další upřesnění odhadu rizika a definice požadavků a vlastností kvality,
- určení klíčových požadavků pro 80% funkčních požadavků,
- vytvoření přesného plánu konstrukční fáze,
- formulace nabídky zahrnující všechny zdroje – prostředky, čas, vybavení, personál a náklady,

Milník fáze rozpracování:

- vytvoření robustního spustitelného architektonického základu,
- je evidován odhad rizik,
- byl vytvořen plán projektu do takové hloubky, aby umožnil vytvoření nabídky,
- obchodní případ byl porovnán s plánem,
- uživatelé odsouhlasili pokračování. [6]

4.1.2. UML diagramy

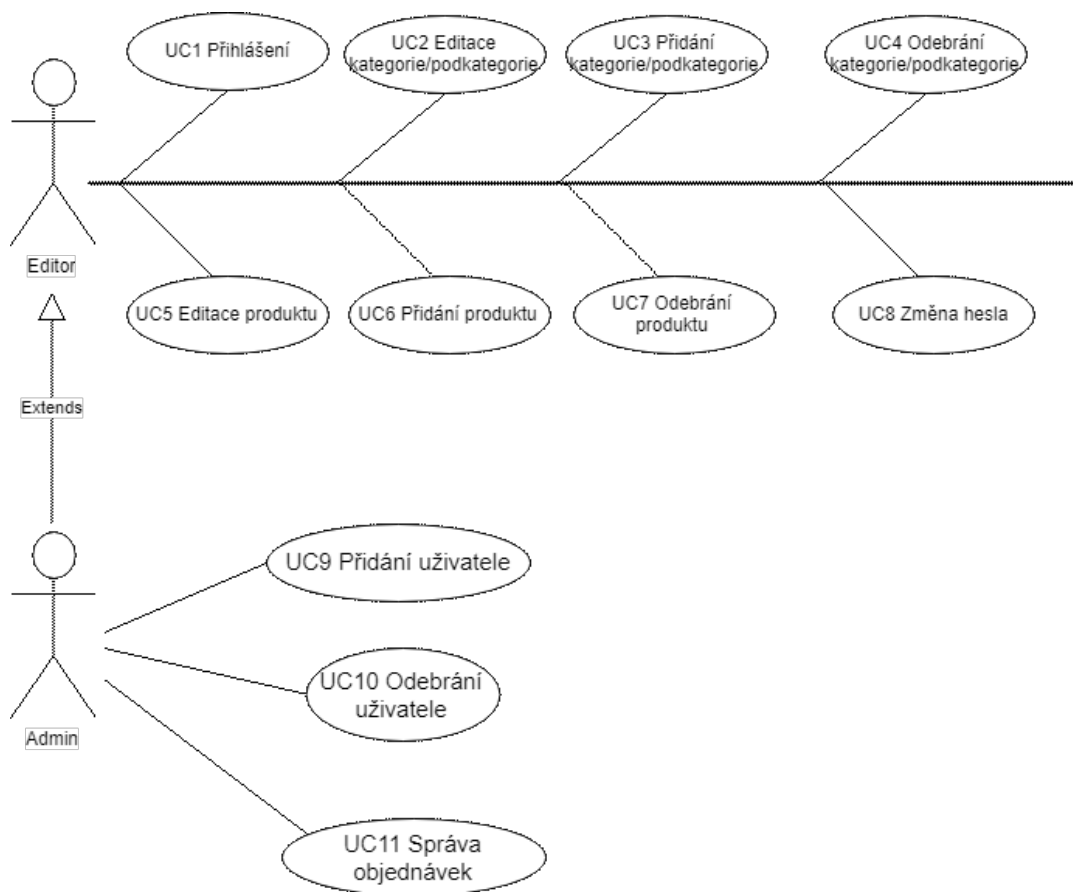
Diagram UML je grafické schéma založené na UML (Unified Modeling Language) s cílem vizuálně reprezentovat systém spolu s jeho hlavními aktéry, rolemi, akcemi, artefakty nebo třídami, s cílem lépe porozumět, měnit, udržovat nebo dokumentovat informace o systému. [18]

4.1.2.1. Případy užití (USE-CASE diagram)

V této kapitole se zaměříme na případy užití z pohledů tří hlavních aktérů:

1. Editor,
2. Admin,
3. User.

Editor + Admin

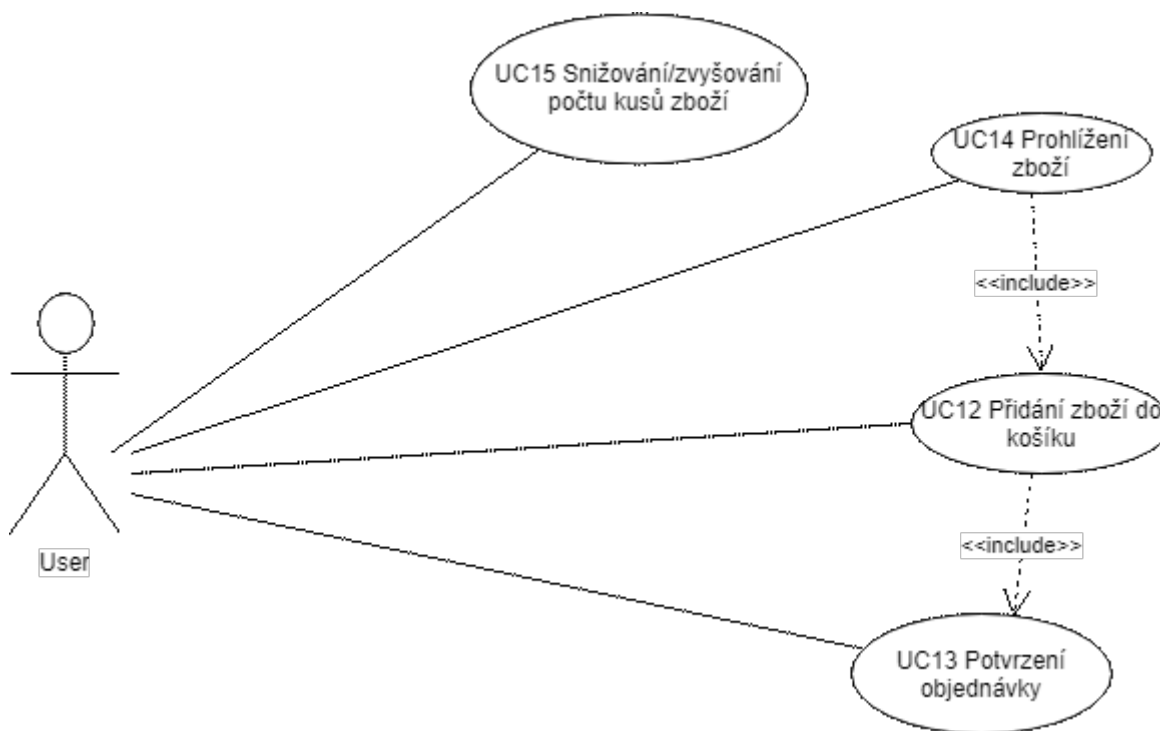


Obrázek 4.4: Use-case diagram: Admin, Editor [vlastní zdroj]

Diagram užití vyobrazený výše slouží k přehlednému grafickému vyjádření struktury, počtu a názvu případů užití, které charakterizují popis chování systému za dané situace (v rámci daného scénáře). V našem diagramu je také za použití vazby Generalizace (dědičnosti) vyjádřena vazba Admina vůči Editorovi, která charakterizuje dědičnost veškerých případů užití od Editora

a k tomu přidává, neboli rozšiřuje, možnosti případů užití v rámci Admina o tři nové a těmi jsou: UC9 – Přidání uživatele, UC 10 – Odebrání uživatele, UC11 – Správa objednávek. Žádný z případů užití není závislý na konkrétní úloze jiného procesu, nicméně veškeré procesy jsou závislé na naší kořenové databázi.

User



Obrázek 4.5: Use-case diagram: User [vlastní zdroj]

Uživatel neboli User v sobě obsahuje čtyři případy užití UC11 – UC14. Z našeho diagramu je patrné, že UC11 Prohlížení zboží je nutným vstupním procesem do UC13 – Přidání zboží do košíku a tento případ užití je vyžadován z toho důvodu, aby byla dostupná možnost UC4 – Potvrzení objednávky zákazníkem.

4.1.2.2.Karty případů užití

Z důvodu, že jsou charaktery některých případů užití podobné a jejich dokumentace je rozsáhlá, nebudou zde uvedeny veškeré karty případů užití a související diagramy. Karty vybraných případů jsou obsaženy v rámci přílohy č. 1.

4.1.2.2.1. Sekce Kategorie

Spravování kategorií (Men, Women, Boys, Girls) bude probíhat skrze administrátorskou část webu, do které se daný uživatel – admin, editor bude moci přihlásit skrze přihlašovací

formulář. Uživateli bude umožněno kategorii či její podkategorie(Shirts, Pants, Hoodies..) modifikovat dle následujících případů užití:

UC3 – Přidání kategorie/podkategorie zboží

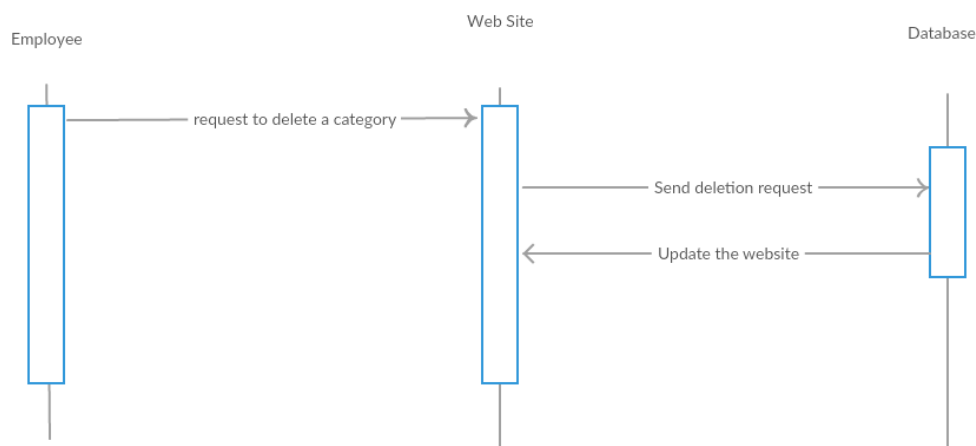
Pomocí jednoduššího formuláře, kde si uživatel bude moci pomocí roletky vybrat rodiče, to v našem případě znamená nadřazená kategorie – Men, Women, Boys, Girls. Na základě vybrané kategorie zde bude k dispozici možnost přidání podkategorie zboží př. Shirts, Pants, Hoodies. Ošetřena bude samozřejmě možnost přidání duplicitního záznamu do databáze, tzn. duplicitní výskyt podkategorie Shoes v sekci Men. Stejným způsobem bude ošetřena také možnost přidání nové kategorie rodiče do databáze.

UC2 – Editace již existující kategorie, podkategorie

Funkcionalita bude dostupná skrze zobrazení seznamu všech kategorií zboží. Vedle názvu kategorie v prvním sloupci, zde bude také sloupec druhý, který bude obsahovat název rodiče = kategorie zboží. Použitím intuitivní vektorové ikony si uživatel bude moci v rámci každého řádku, tedy každé jednotlivé položky kategorie, zvolit pomocí kliknutí na ikonu tužky pro editaci zvolené kategorie/podkategorie, kdy mu bude umožněno přepsat název požadované položky.

UC4 – Odebrání kategorie/podkategorie

Tato funkcionalita bude uživateli k dispozici opět pomocí užití vektorové ikony, nyní však ikony koše, která bude umístěna vedle ikony tužky pro editaci. Sekvenční diagram popisuje způsob, jakým tato funkcionalita bude provozována. Uživatel, v našem případě zaměstnanec, prostřednictvím webové rozhraní vybere možnost odebrání kategorie, požadavek je poté odeslán k provedení do databáze a výsledkem je poté obnovení stránky obsahující seznam nyní již bez vybrané položky.

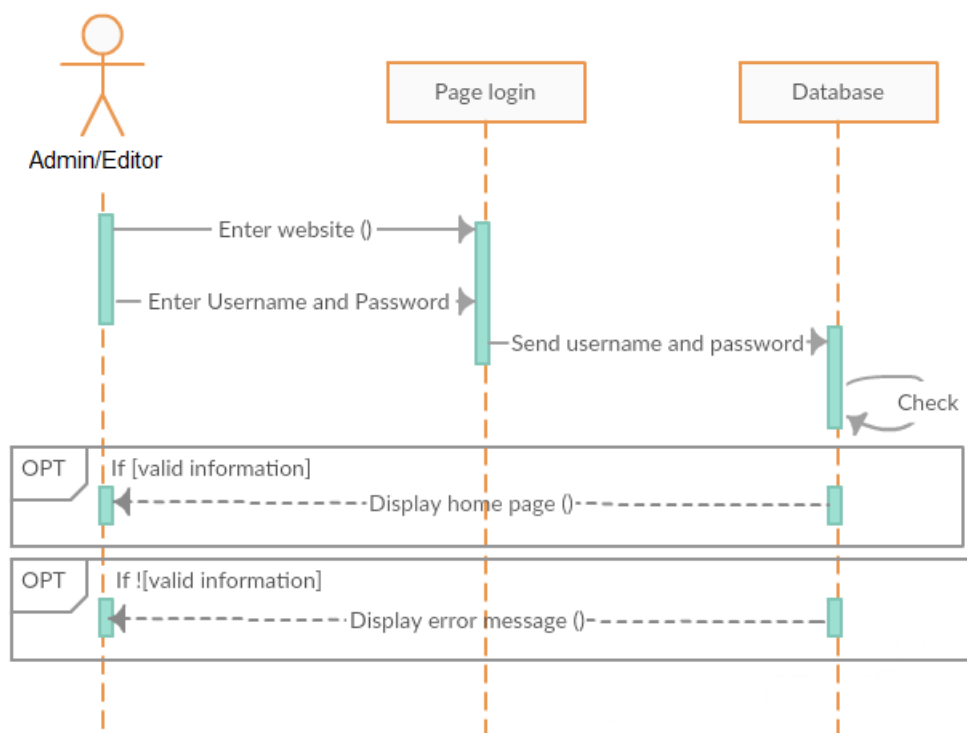


Obrázek 4.6: Sequence diagram: UC4 [vlastní zdroj]

Všechny tři případy užití budou uživateli k dispozici skrze uživatelské webové rozhraní.

4.1.2.2.2. UC1 – Přihlášení

V příloze číslo 1. nalezneme scénář případu užití č.1, který se věnuje procesu přihlašování administrátora či editora do části správy webu. Způsob, kterým bude přihlašovací proces implementován, je zobrazen v rámci sekvenčního diagramu. Tento typ diagramu je užíván zejména v případě, kdy je důležité znázornit časovou souvislost jednotlivých interakcí mezi objekty. Možnost přihlášení do administrátorské části webu bude mít pouze sám správce celého e-shopu a editor, který však bude mít omezené funkcionality. Diagram obsahuje jednoho aktéra, kterým je Admin, případně Editor. Dále zde vidíme dva objekty, které symbolizují přihlašovací stránku a databázovou část. Admin/editor nejprve posílá dvě zprávy – první charakterizuje vstup na webovou stránku a druhá zadání uživatelského jména a hesla. Tento požadavek je následně prostřednictvím HTTP požadavku poslán na server a do databáze, kde je následně provedena verifikace údajů. V případě, že přihlašovací jméno i heslo souhlasí uživateli je povolen přístup do admin. části. V případě, že jsou jedna nebo obě hodnoty zadány chybně, následuje zobrazení chybového hlášení s podrobným popisem chyby.



Obrázek 4.7: Sequence diagram:UC1 [vlastní zdroj]

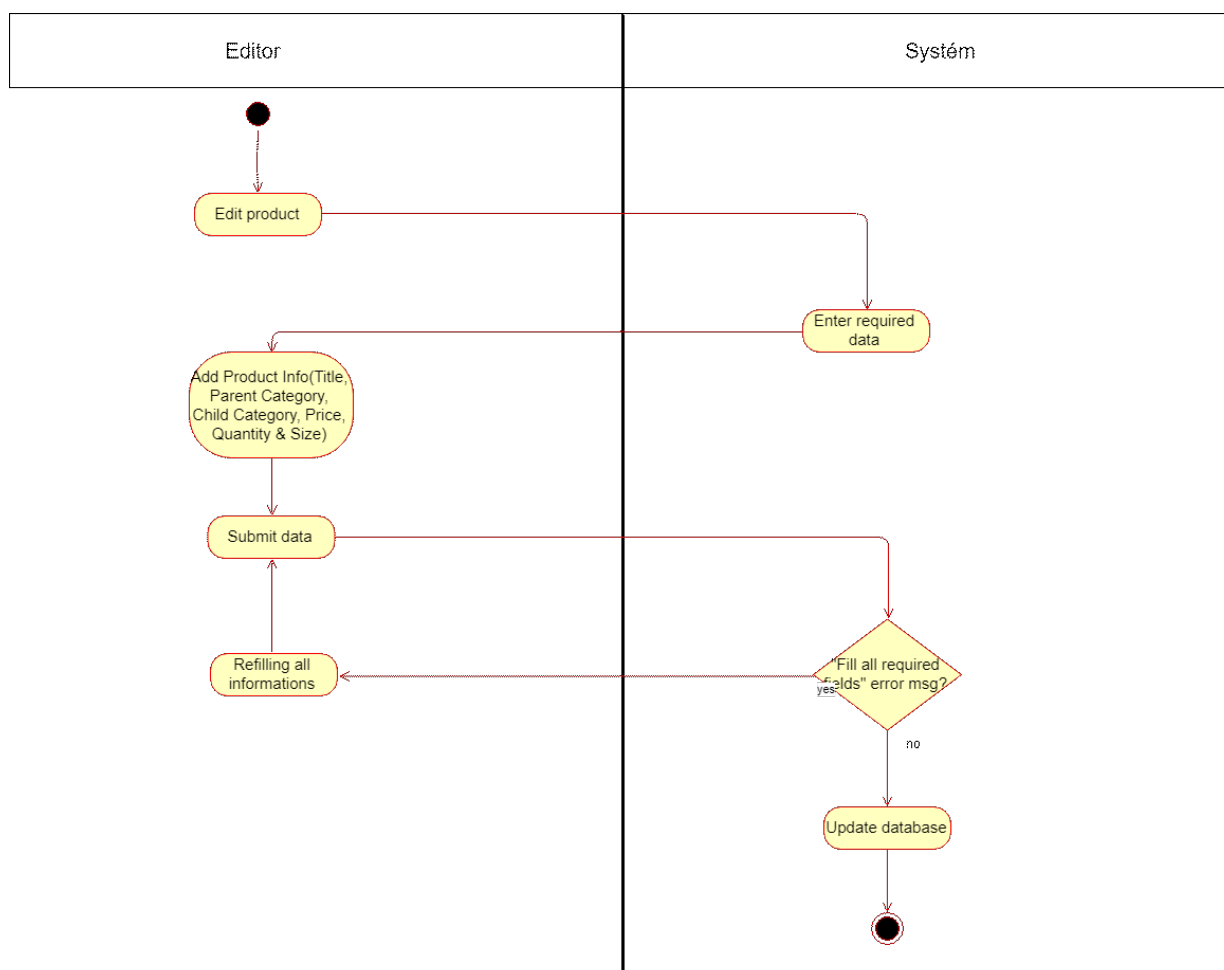
4.1.2.2.3. Sekce Produkty

UC6 – Přidání produktu do databáze

Kliknutím na tlačítko „Add product“ umístěné v sekci Products v části správy webu nám systém zobrazí formulář s poli, které budou blíže popsány v rámci UC5 – Editace produktu. Za předpokladu, že veškeré položky pole budou vyplněny správně, dojde k přidání produktu do databáze. V opačném případě bude systémem vyvoláno chybové hlášení s popisem chyby.

UC5 – Editace produktu

Položka produktu bude obsahovat tyto informace: Title, Parent Category, Child Category, Price, List Price, Quantity & Size, Description, Product Photo. Povinné vlastnosti daného produktu budou vyznačeny znakem hvězdy za názvem dané vlastnosti např. Title*. Mezi nepovinná pole produktu bude patřit: List Price, která charakterizuje běžnou cenu v ostatních obchodech; Description a Product photo. Stejně tak jako funkcionality přidání a odebrání produktu, také tato funkcionality jsou dostupné jak v případě administrátorských, tak i editorských přístupových práv. Užitečnost funkcionality si popíšeme pomocí modelové situace. Velké množství firem si v současné době v rámci dobrého jména a snahy zaujmout zákazníka fotí své produkty samy.



Obrázek 4.8: Activity diagram:UC5 [vlastní zdroj]

V případě, že se pro prezentování zboží prostřednictvím vlastních fotek nerozhodnete hned ze začátku publikování zboží na Internetu, díky možnosti editace stávajícího produktu odpadá nutnost vytváření nového produktu a překopírování veškerých dat. Dalšími modelovými příklady mohou být také změna ceny nebo úprava popisku zboží v rámci aplikace pravidel pro SEO. I v tomto případě je validita vstupních dat ošetřena ze strany systému případným zobrazením chybového hlášení. Celkový seznam produktů nalezneme na úvodní stránce Products ve správcovské části webu, kde bude také dostupná možnost výběru produktů, které budeme chtít zobrazit na hlavní stránce webu, např. nové produkty, nejčastěji prodávané produkty.

UC7 – Odebrání produktu

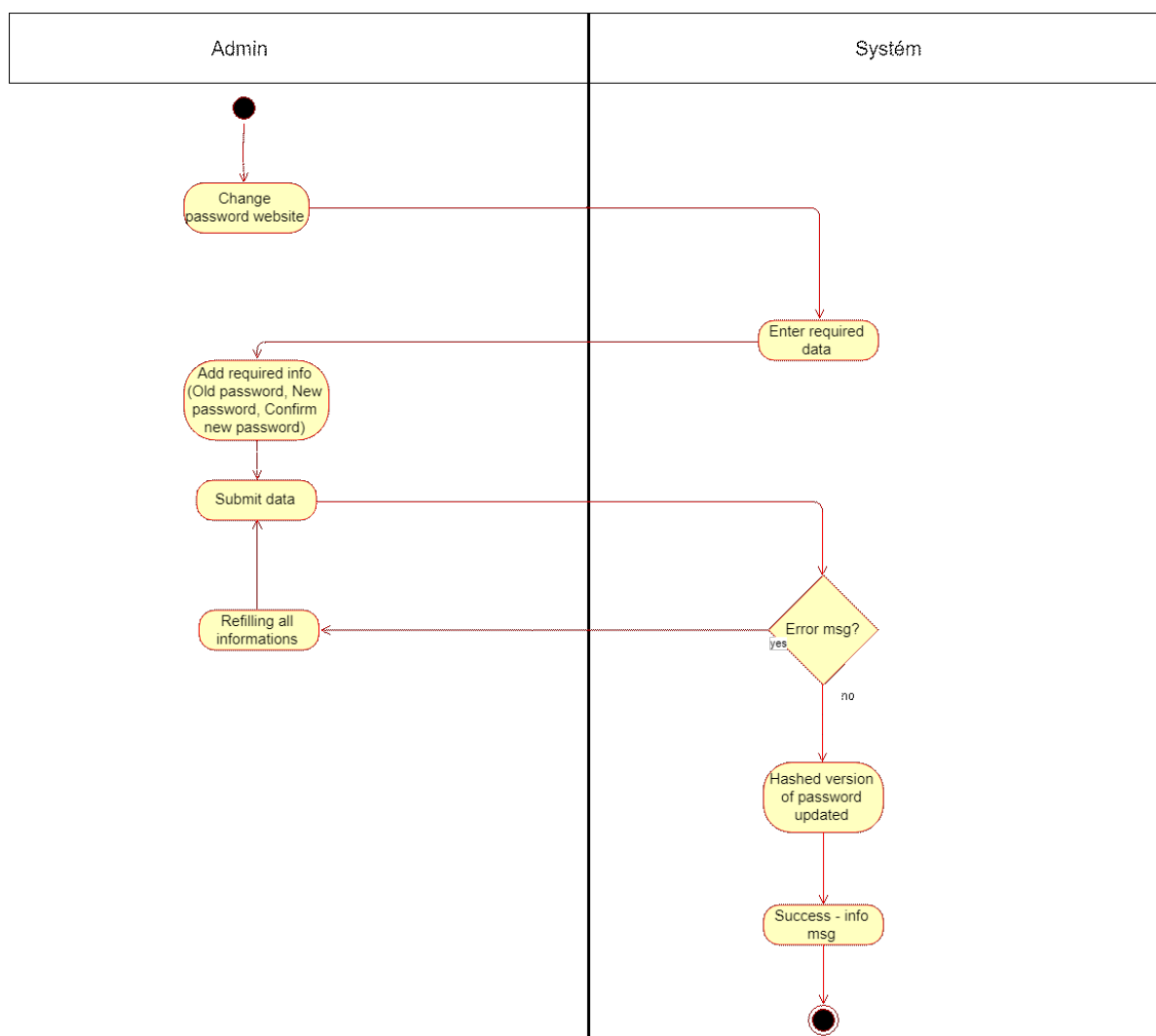
I v tomto případě je možnost odebrání produktu prezentována uživateli použitím vektorové ikony koše po levé straně, hned vedle ikony tužky, pro úpravu produktu. Požadavek o smazání produktu, který identifikujeme dle systémem přiřazeným ID bude odeslán do naší

databáze a poté dojde k aktualizaci stránky, obsahující seznam produktů již bez námi zvoleného produktu

4.1.2.2.4. Uživatelská sekce

UC8 – Změna hesla

Změnu hesla bude opět moci provádět pouze Admin, nikoliv Editor. V případě nutnosti změny hesla bude k dispozici funkcionality určená k provedení tohoto požadavku. Algoritmus je velice podobný jako většiny ostatních případů užití. Požadavek vychází od administrátora, načež systém požaduje zadání informací nezbytně nutných k úspěšnému provedení tohoto požadavku. V případě, že uživatel zadá korektně veškeré hodnoty, systém aktualizuje heslo v databázi a správci webu zobrazí oznámení, zdali požadavek proběhl v pořádku.

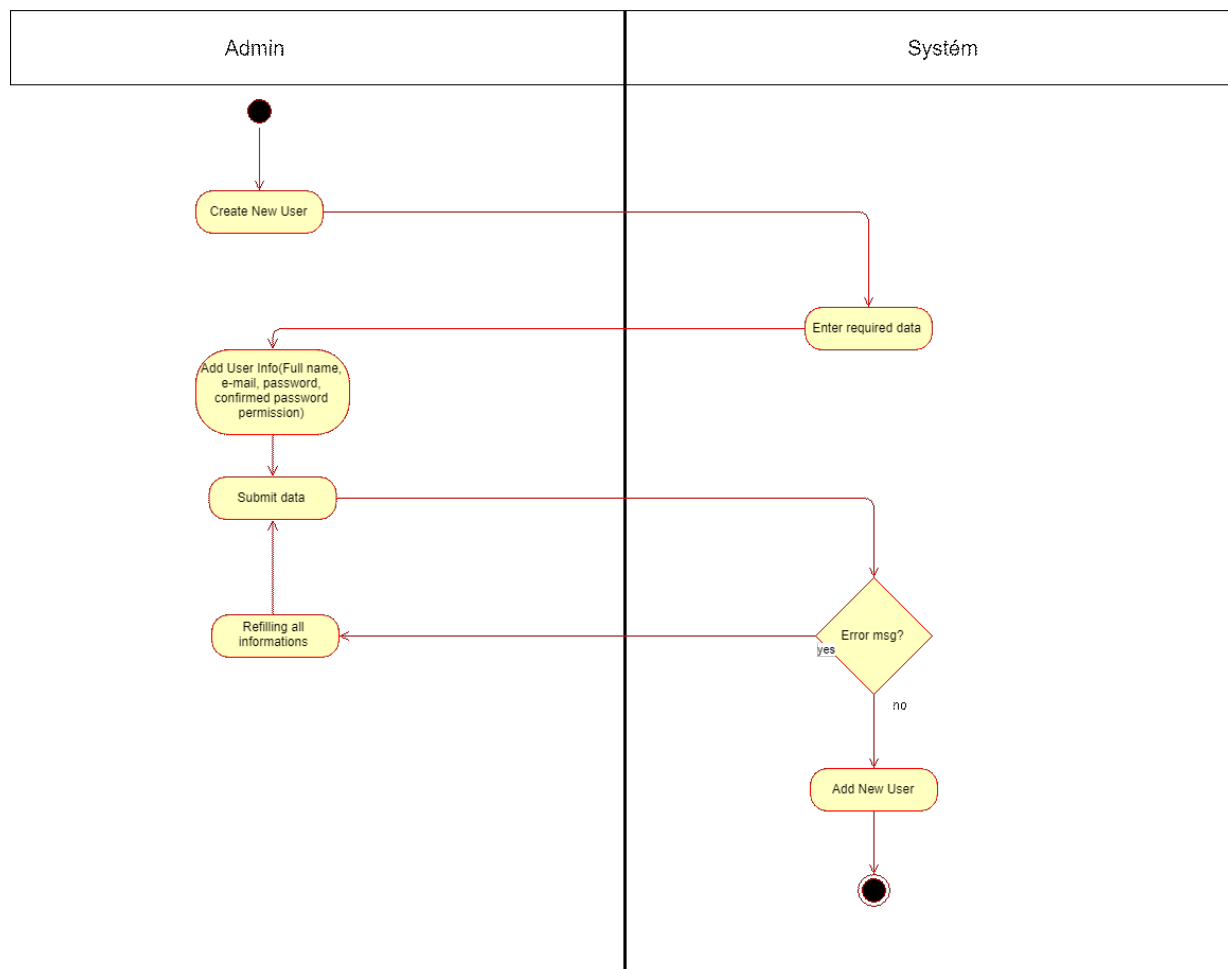


Obrázek 4.9: Activity diagram:UC8 [vlastní zdroj]

UC9 – Přidání uživatele

Proces přidání uživatele si nejlépe znázorníme za použití Activity diagramu a jeho metody plavebních drah, která se využívá pro rozdělení diagramů na více částí za účelem názornějšího vykreslení odpovědnosti jednotlivých účastníků v daném procesu za různé části aktivity. Diagram obsahuje dva základní pohledy, a tím je pohled Admina a poté zachycení odezvy systému. První akce přichází logicky ze strany Admina, který posílá požadavek na vytvoření nového uživatele. Systém na tuto akci reaguje požadováním vyplnění vstupních údajů nutných pro úspěšné přidání uživatele. Admin na vyžádání doplňuje potřebná data a potvrzuje jejich správnost odesláním.

Systém nejprve zkontroluje, zdali je tvrzení, že jsou opravdu všechna data vyplněna pravdivé – v případě, že ano, přidá nového uživatele do databáze; v případě, že ne, je Admin povinen data doplnit a opět je zaslat zpět. Opět probíhá kontrola správnosti údajů do té doby, než jsou veškerá data správně vyplněna.

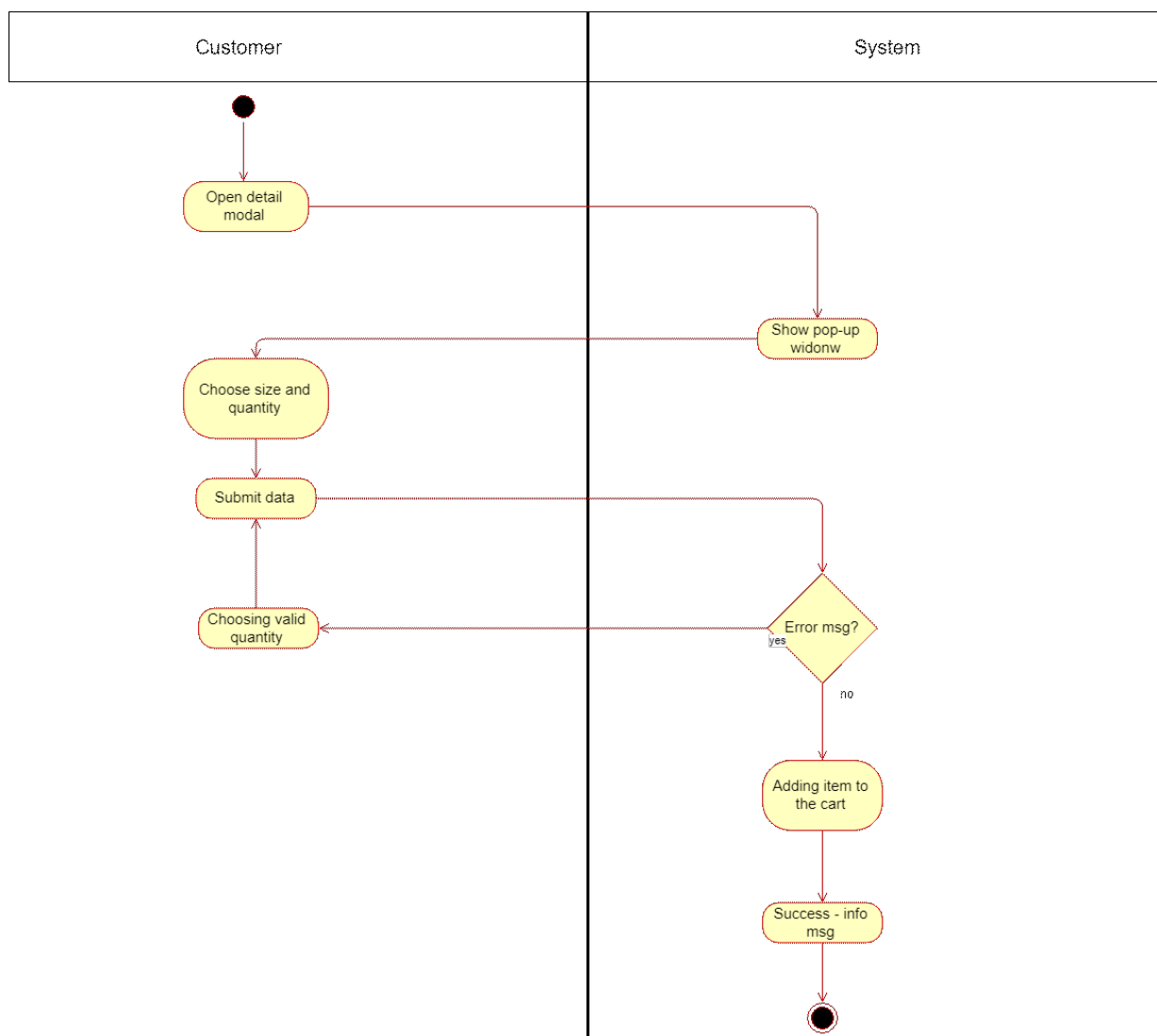


Obrázek 4.10: Activity diagram: UC9 [Interní zdroj]

4.1.2.2.5. Objednávkový proces

UC12 Přidání zboží do košíku

K přidání konkrétní položky do košíku je nejprve nutné kliknutí na tlačítko „Details“, které je umístěné pod každou položkou. Otevře se nám vyskakovací okno, ve kterém uvidíme detailnější popis daného produktu a také možnost v podobě tlačítka „Add to cart“. Aby byla položka úspěšně přidána do košíku, je nutný výběr velikosti a korektního množství kusů daného zboží. Po potvrzení formuláře mohou nastat dvě situace. Situace první charakterizuje validní vstupní data, což vede k úspěšnému přidání daného zboží do nákupního košíku. Situace druhá nastane tehdy, jestliže zákazník vybere větší počet kusů než je momentálně dostupný na skladě – aktuální počet kusů, který je dostupný je vždy uveden v závorce u dané velikosti. V tomto případě je vyhozena chybová zpráva a přidání položky do košíku nebude provedeno.



Obrázek 4.11: Activity diagram:UC12 [vlastní zdroj]

4.1.3. Databázový systém

Databázový systém je spravován pomocí softwarového nástroje phpMyAdmin založený na MySQL databázi. Systém je složen ze čtyř základních tabulek:

4.1.3.1.1. Users

#	Name	Type	Collation	Default	Other
1	id(PK)	int(11)			AUTO_INCREMENT
2	full_name	varchar(255)	utf8_unicode_ci		
3	email	varchar(175)	utf8_unicode_ci		
4	password	varchar(255)	utf8_unicode_ci		
5	join_date	datetime			
6	last_login	datetime		CURRENT_TIMESTAMP	
7	permission	varchar(255)	utf8_unicode_ci		

Tabulka je složena ze sedmi sloupců/atributů tabulky:

1. id – atribut id je zároveň primárním klíčem tabulky, datový typ je int o délce 11 bytů, který se užívá pro celočíselné hodnoty, tento atribut slouží k jednoznačné identifikaci uživatele ve formě identifikačního čísla
2. full_name – atribut full_name reprezentuje jméno a příjmení uživatele, datový typ je varchar o délce 255 bytů
3. email – reprezentuje e-mail uživatele, datovým typem je opět varchar, nicméně pro tento atribut je vyhrazeno pouze 175 bytů
4. password – reprezentuje heslo uživatele, datový typ varchar o délce 255 bytů
5. join_date – datum, kdy byl uživatel přidán do databáze, datový typ datetime
6. last_login – datum posledního přihlášení uživatele, datový typ datetime, výchozí hodnotou je hodnota aktuálního data a času
7. permission – přístupová práva uživatele, může nabývat dvou hodnot: Admin, Editor, základní rozdělení z důvodu omezení přístupu do všech sekcí v případě editora, datový typ je varchar o délce 255 bytů

4.1.3.1.2. Products

#	Name	Type	Collation	Default	Other
1	id(PK)	int(11)			AUTO_INCREMENT
2	Title	varchar(255)	utf8_unicode_ci		
3	Price	decimal(10,2)			
4	list_price	decimal(10,2)			
5	categories	varchar(255)	utf8_unicode_ci		
6	Image	varchar(255)	utf8_unicode_ci		
7	description	Text	utf8_unicode_ci		
8	featured	tinyint(4)		0	
9	Sizes	Text	utf8_unicode_ci		

10	Deleted	tinyint(4)		0	
----	---------	------------	--	---	--

Tabulka je složena z deseti sloupců/atributů tabulky:

1. id – atribut id je zároveň primárním klíčem tabulky, datový typ je int o délce 11, slouží k identifikaci daného produktu
2. title – atribut title reprezentuje název produktu, datový typ je varchar o délce 255 bytů
3. price – reprezentuje cenu produktu, datovým typem je decimal o délce 10 a se zaokrouhlením na dvě desetinná místa
4. list_price – reprezentuje katalogovou cenu produktu, datový typ je decimal
5. categories – reprezentuje druh zboží (shirts, hoodies..), datový typ je varchar o délce 255 bytů
6. image – slouží pro uložení absolutní cesty k fotce produktu, datový typ je varchar o délce 255 bytů
7. description – popis produktu, datový typ text
8. featured – pro identifikování produktů, které budou zobrazeny na hlavní stránce, datový typ tinyint o délce 4
9. sizes – dostupné velikosti zboží, datový typ text (např. triko bude mít velikost S, nicméně obuv bude očíslována pomocí číslice)
10. deleted – identifikace smazaných produktů, datový typ tinyint o délce 4

4.1.3.1.3. Categories

#	Name	Type	Collation	Default	Other
1	id(PK)	int(11)			AUTO_INCREMENT
2	category	varchar(255)	utf8_unicode_ci		
3	parent	int(11)		0	

Tabulka je složena ze tří sloupců/atributů tabulky:

1. id – atribut id je zároveň primárním klíčem tabulky, datový typ je int o délce 11 bytů
2. category – atribut category reprezentuje druhy nabízeného zboží, datový typ je varchar o délce 255 bytů
3. parent – reprezentuje kategorií dle které jsou dostupné určité druhy zboží (Men, Women, Boys, Girls), datovým typem je int o délce 11 bytů

4.1.3.1.4. Cart

#	Name	Type	Collation	Default	Other
1	id(PK)	int(11)			AUTO_INCREMENT
2	items	text	utf8_unicode_ci		
3	expire_date	datetime			
4	paid	tinyint(4)		0	
5	shipped	tinyint(4)		0	

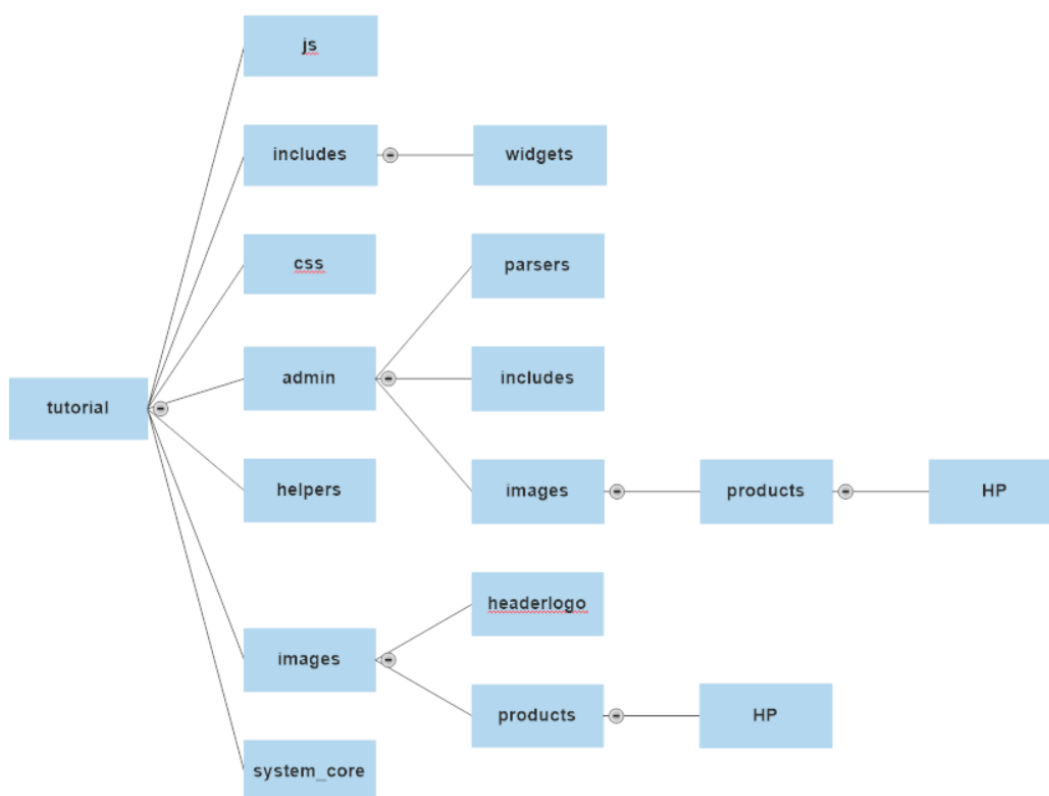
Tabulka je složena z pěti sloupců/atributů tabulky:

1. id – atribut id je zároveň primárním klíčem tabulky, datový typ je int o délce 11 bytů, slouží k jednoznačné identifikaci nákupu
2. items – atribut items reprezentuje položky košíku, datový typ je text
3. expire_date – reprezentuje datum vypršení cookie pro daného uživatele, datovým typem je datetime
4. paid – reprezentuje stav objednávky, kdy byla objednávka zaplacená, datovým typem je tinyint o délce 4 bytů
5. shipped – reprezentuje stav objednávky, kdy byla odeslána, datovým typem je opět tinyint o délce 4 bytů

4.1.3.1.5. Návrh systémové části

Stromová struktura projektu

K lepší orientaci v projektu jako takovém, složkách a částech projektu software Atom nabízí v rámci podélného levého panelu možnost zobrazení stromové struktury aktuálně otevřeného projektu.



Obrázek 4.12: Stromová struktura projektu [Interní zdroj]

Vrcholem stromu našeho zdrojového kódu je složka tutorial, ve které se nachází základní soubory použité pro implementaci uživatelského rozhraní. Tento hlavní adresář se dále větví na složky:

- css – složka obsahující kořenový stylpis knihovny Bootstrap,

- js – složka obsahující implementaci souborů JS a jQuery, součástí knihovny Bootstrap,
- images – obsahující logo a produktové fotky,
- system_core – obsahuje klíčový soubor init.php, ve kterém je nadefinováno připojení k databázi a také práce s relacemi (Sessions),
- includes – složka obsahuje osm .php dokumentů, které obsahují stavební bloky e-shopu např. navigace, detailní modul produktu apod.; nachází se zde také ještě jedna další složka s názvem widget, ve které nalezneme dva soubory, jeden obsahující implementaci aplikování filtru při vyhledávání zboží a druhý pro práci s košíkem,
- helpers – soubor helpers.php je zdrojem metod, které se používají v rámci více částí e-shopu jako např. JS funkce `display_errors($errors)`, která již podle názvu zajišťuje manipulaci s chybami, které mohou nastat v důsledku nesprávného použití systému,
- admin – zahrnuje soubory obsahující zdrojový kód pro administrační část webu jako seznam uživatelů, sekci určenou pro editaci kategorií, či produktů. Také zde nalezneme složku includes, která v sobě opět obsahuje stavební bloky administrační části a složku parsers, která obsahuje implementaci nákupního formuláře.

4.1.4. Návrh vzhledu

Univerzální odpověď na otázku, jaký je správný postup při návrhu designu prostě a jednoduše neexistuje. Každý projekt si klade jiné cíle a má odlišné časová, finanční nebo technická omezení. Nicméně stále platí, že vzhled internetového obchodu je jednou z nejdůležitějších částí v procesu návrhu elektronického obchodu. Návrh rozložení prvků webové stránky byl zhotoven v podobě Wireframu a barevné schéma bylo navrženo prostřednictvím webového editoru Adobe Color Wheel.

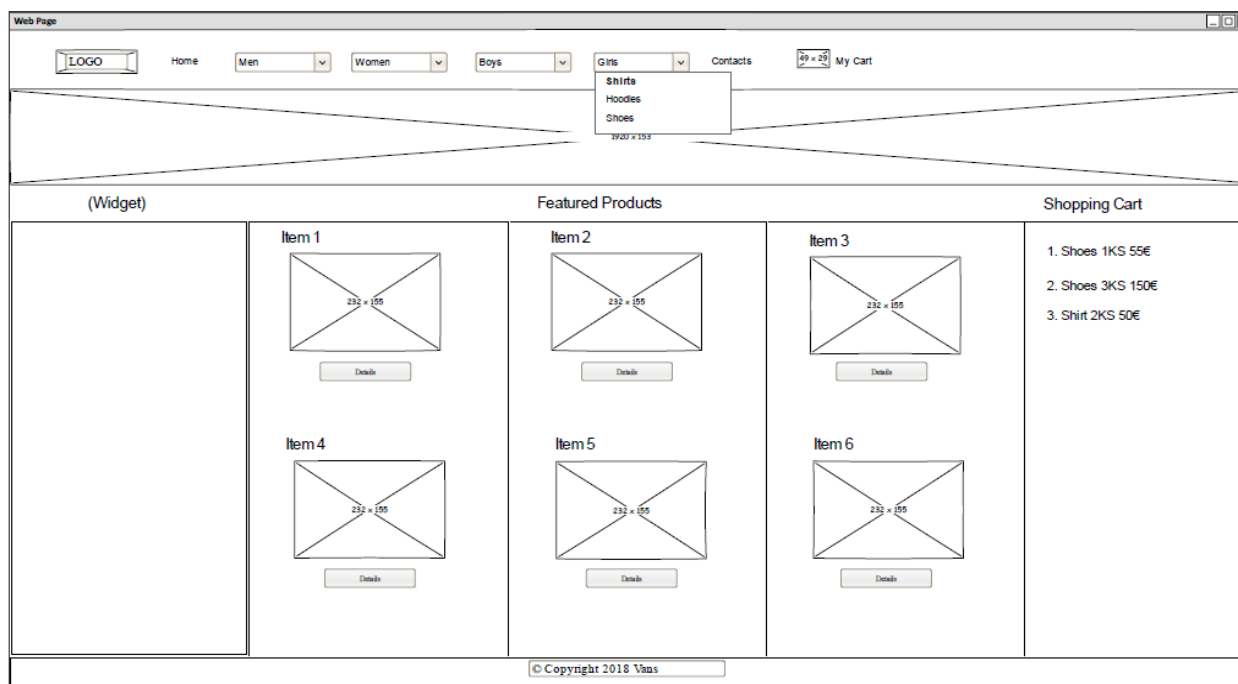
Rozmístění hlavních prvků internetového obchodu se drží standardních formálních internetových zvyklostí (odkaz na přihlašovací formulář vpravo nahoře, navigační menu s logem značky vlevo nahoře apod.). Díky zachování těchto zvyklostí se zákazník rychle a snadno orientuje po celém e-shopu.

4.1.4.1. Wireframe

Wireframe neboli drátěný model webu se běžně používá v oblasti vývoje webu, či e-shopu v druhé vývojové fázi, kterou nazýváme „Návrh webu“ a jak už název napovídá, jedná se o tzv. kostru webové prezentace či aplikace pro náhled nového řešení. Účelem drátěného modelu

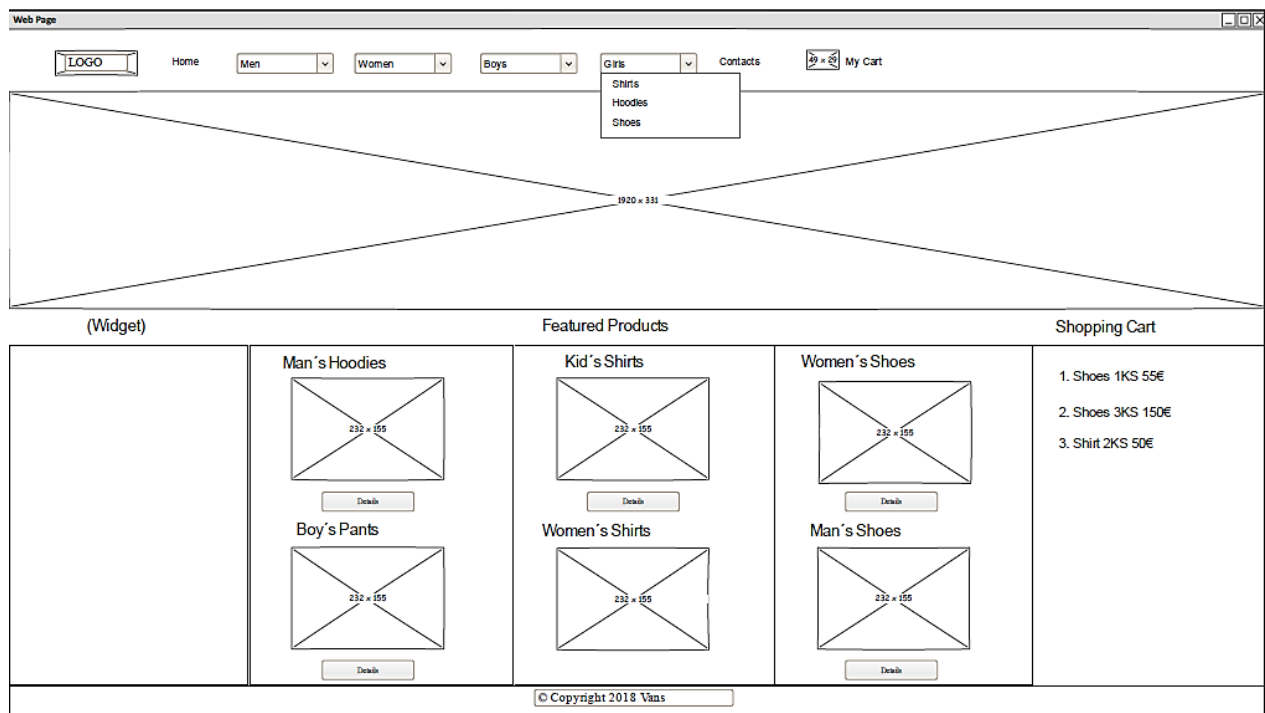
je definice funkce webu a obsahu webu, což zahrnuje aktivity jako rozmístění funkčních prvků na stránce. Ačkoliv se na první pohled může zdát, že se jedná o grafický návrh webu, není tomu tak. Wireframe neobsahuje konkrétní obrázky a je tvořen pouze pomocí stanovených nejčastěji jednobarevných objektů, případně textu. Z hotového řešení lze pak vyčíst, co se kde nachází a vzájemné propojení objektů, z čehož následně vyplývá již zmíněná funkce a rozmístění obsahu na webu.

4.1.4.1.1. Úvodní stránka



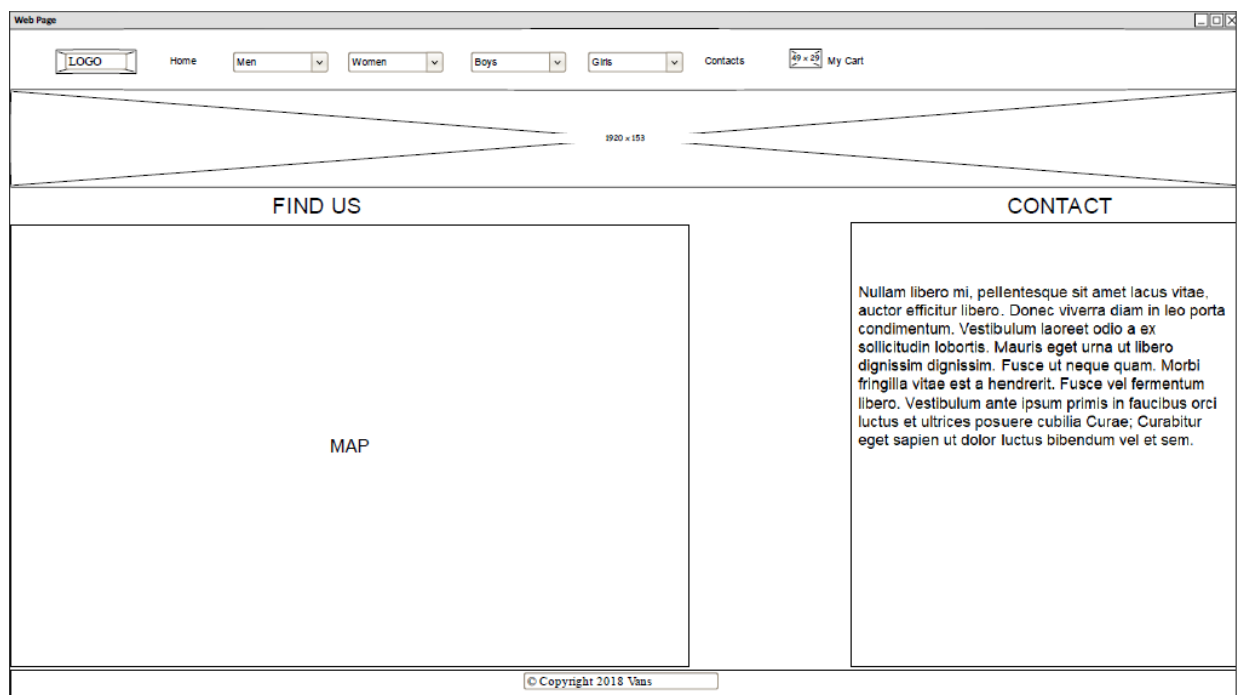
Obrázek 4.13: Wireframe – Úvodní stránka [Interní zdroj]

4.1.4.1.2. Detail položky



Obrázek 4.14: Wireframe – Detailní náhled kategorie [Interní zdroj]

4.1.4.1.3. Kontakty



Obrázek 4.15: Wireframe – Kontakty [Interní zdroj]

4.1.4.1.4. Košík

My Shopping Cart					
#	Item	Price	Quantity	Size	Sub Total
1	Retro sun boyfriend tee	\$29.50	2	S	\$59.00
Totals					
Total item		Sub Total	Tax	Grand Total	
2		\$59.00	\$12.39	\$71.39	
Check Out					
© Copyright 2018 Vaus					

Obrázek 4.16: Wireframe – Košík [Interní zdroj]

4.1.5. Použitý software

K návrhu, vývoji a implementaci elektronického obchodu byla zapotřebí následující softwarová řešení:

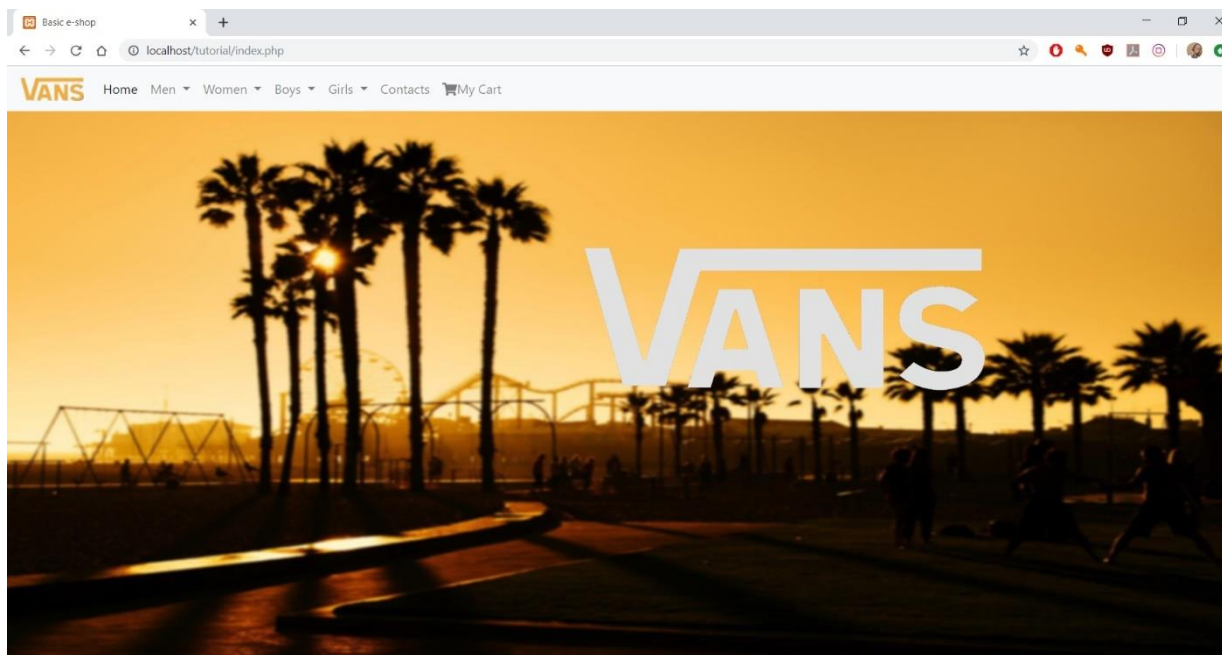
- software pro správu kódu: Atom,
- software pro správu databáze: phpMyAdmin,
- webový server pro testování: XAMPP,
- webový prohlížeč pro testování: Google Chrome, Edge, Firefox.

4.2. Implementace

4.2.1. Uživatelské rozhraní

4.2.1.1. Úvodní stránka

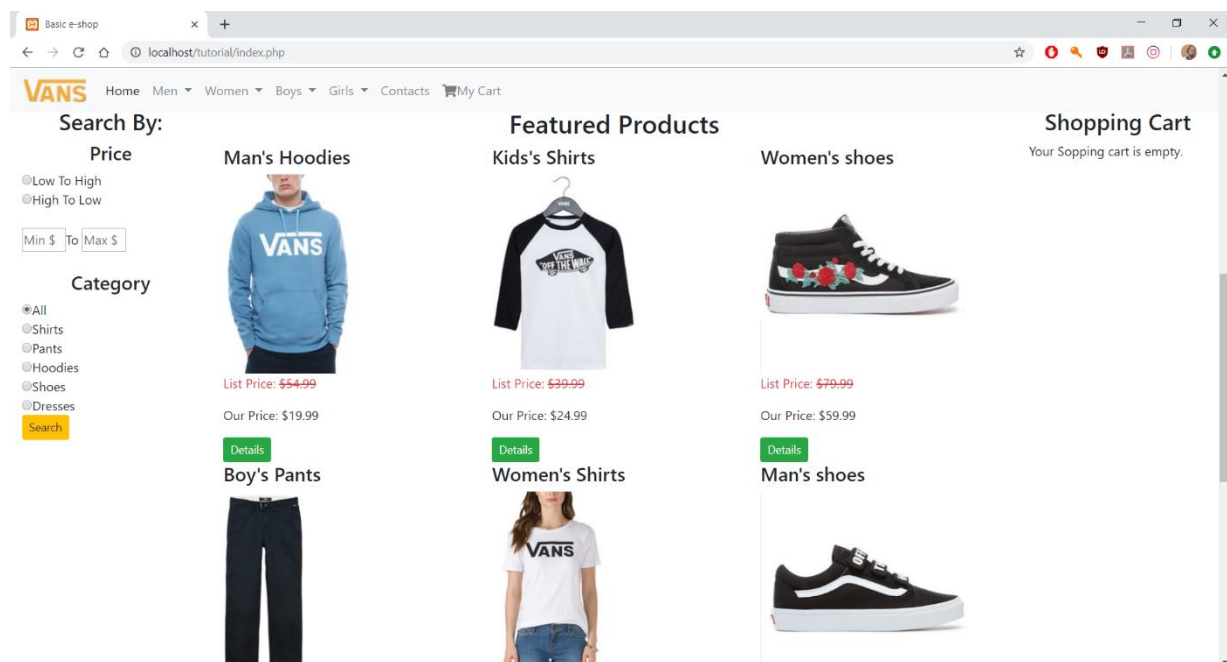
Home Page neboli Úvodní stránka je tvořena v horní části logem firmy a navigačním panelem, obsahujícím seznam odkazů, sekcí, které jsou návštěvníkovi webu k dispozici. Menu(navigační panel) sestává ze sedmi odkazů umístěných vedle sebe – Home, Men, Women, Boys, Girls, Contacts, My Cart. Textový popis odkazu je tvořen až na výjimky Home, Contacts a My Cart tvořen názvy kategorií, dle kterých je dané zboží rozděleno, tzn. Pánská móda, dámská, pro chlapce a děvčata. Součástí menu je v případech těchto kategorií rozbalovací seznam daných druhů zboží. V případě pánské módy – Shirts, pants, hoodies, shoes. Pro dámy zde nabízíme Shirts, Dresses, Hoodies, Shoes. Chlapecký výběr módy sestává z výběru mezi Shirts, pants, hoodies, shoes. A nakonec pro děvčata zde nabízíme Shirts, hoodies, shoes.



Obrázek 4.17: Úvodní stránka [Interní zdroj]

Spodní část úvodní stránky je z velké části tvořena ukázkou vybraného zboží napříč jednotlivými kategoriemi. Produkty jsou umístěny vedle sebe v rámci tzv. grid systému. Každý řádek obsahuje výběr tří produktů a ke každému je zde uvedena fotka produktu, katalogová cena a cena platící v rámci našeho e-shopu. Pod těmito informacemi je umístěno zelené tlačítko s názvem: “Details“, které slouží k otevření modulu, ve kterém si požadovaný produkt můžeme prohlédnout detailněji. Pravý panel je tvořen widgetem, který umožňuje rychlý náhled na

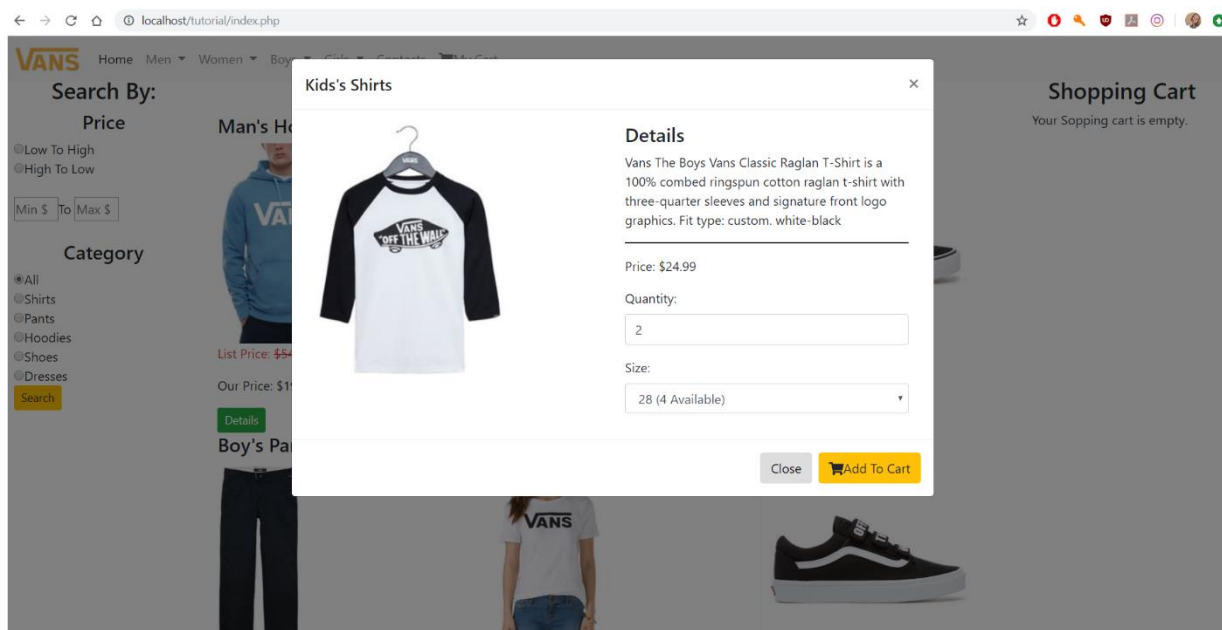
aktuální stav košíku. Levý panel poté nabízí možnost vyhledávání produktů, dle určitých specifikací např. podle ceny ve specifikovaném rozmezí nebo dle kategorie daného zboží – pánská, dámská, chlapecká a dívčí. Výběr filtru, dle kterého chceme produkty hledat potvrdíme stisknutím žlutě zbarveného tlačítka s nápisem: „Search“ umístěného pod seznamem kategorií v rámci stejného panelu.



Obrázek 4.18: Úvodní stránka 2 [Interní zdroj]

4.2.1.2. Detail produktu

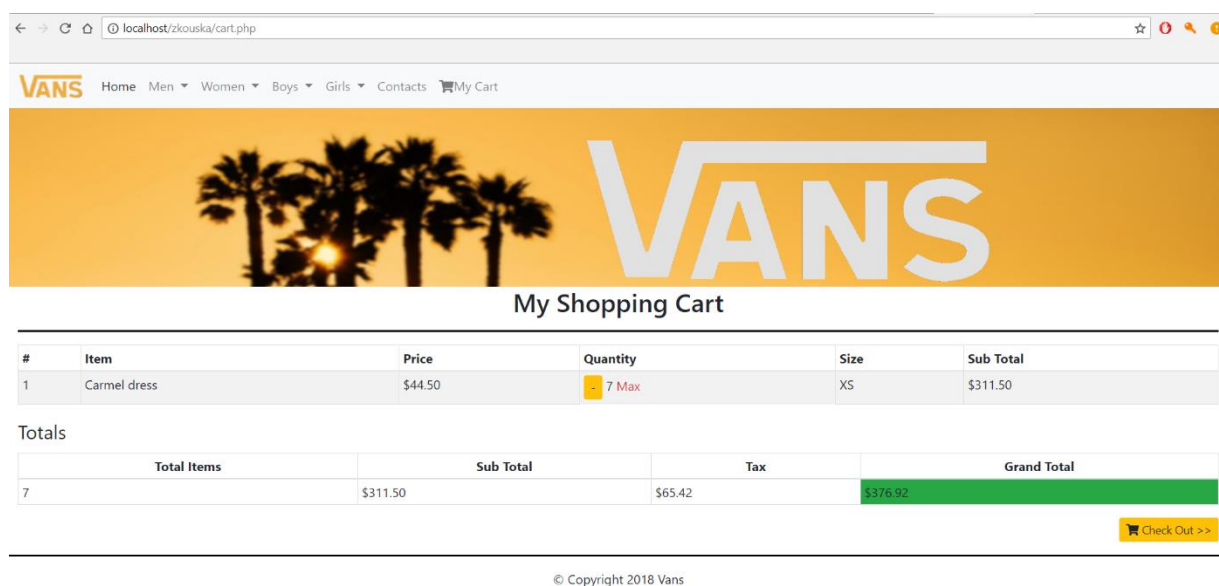
Detail produktu je zde implementován za použití vyskakovacího okna/modulu, které se uživateli zobrazí při kliknutí na zelené tlačítko: „Details“ umístěné v kategorii produktu nebo na hlavní stránce e-shopu. Modul vždy obsahuje informace o názvu daného produktu v levém horním rohu. Nepovinná, nicméně vhodná je také fotka produktu umístěna pod názvem produktu v levé části modulu. Pravá strana obsahuje detailní popis produktu, cenu zboží v dolarech a dvě pole. Vstupní pole s názvem Quantity slouží k zadání počtu kusů zboží, které si zákazník přeje objednat. Druhé pole je implementováno formou rozbalovacího seznamu, kde jsou uvedeny dostupné velikosti a také počet dostupných kusů dané velikosti zboží. Spodní část je složena ze dvou tlačítek – šedé tlačítko s nápisem „Close“ slouží stejně tak, jako ikona křížku v pravém horním rohu k zavření modulu, žluté tlačítko „Add to cart“ je určeno k přidání dané položky do košíku.



Obrázek 4.19: Detail produktu [Interní zdroj]

4.2.1.3. Košík

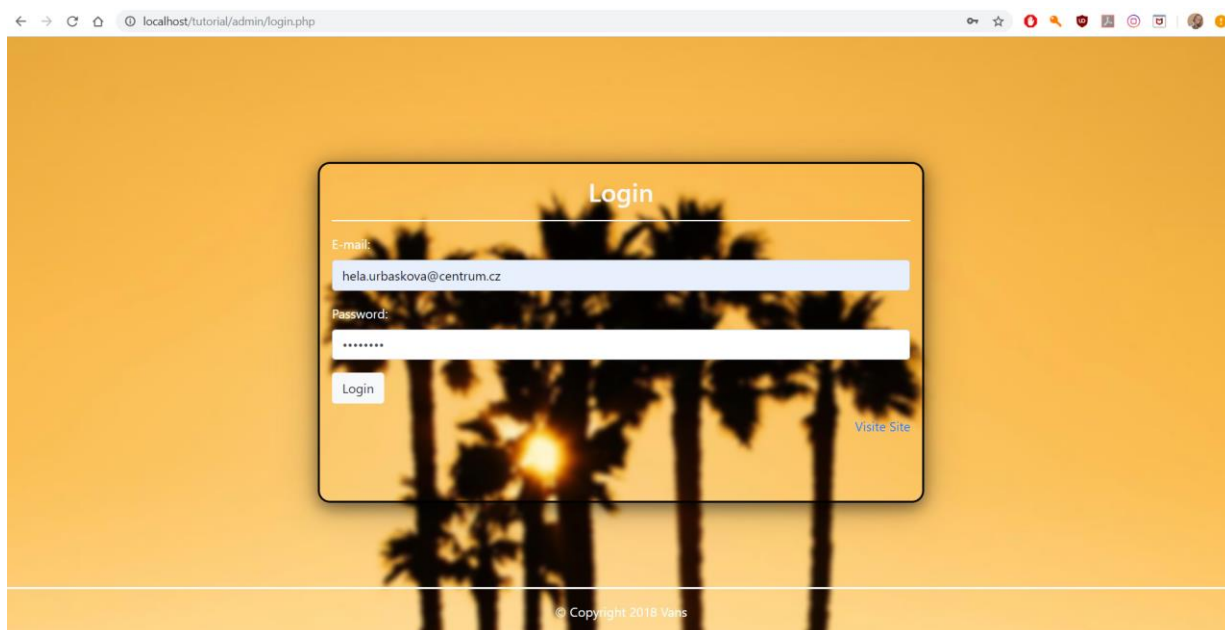
Stránka Košík zákazníkovi nabízí detailní pohled na obsahu košíku v podobě tabulky, která je umístěna pod nadpisem tabulky „My Shopping Cart“. Tabulka obsahuje šest sloupců z nichž první označuje pořadí položky v košíku, druhý sloupec je určen pro název produktu, třetí pro cenu za kus. Čtvrtý k zobrazení k množství kusů zákazníkem vybraného v rámci dané položky zboží. V pátém sloupci se nachází cena za daný počet kusů a sloupec číslo sedm udává cenu za daný počet kusů pro jednotlivou položku. Ve spodní části stránky jsou k dispozici součty počtu položek v košíku, celková výše ceny bez DPH, výše ceny DPH a celková suma objednávky včetně DPH. Pomocí žlutého tlačítka „Check out >>“ se zákazník přesune k dalšímu kroku v procesu objednávky, kterým je vyplnění údajů o adresátovi.



Obrázek 4.20: Košík [Interní zdroj]

4.2.1.4. Přihlášení

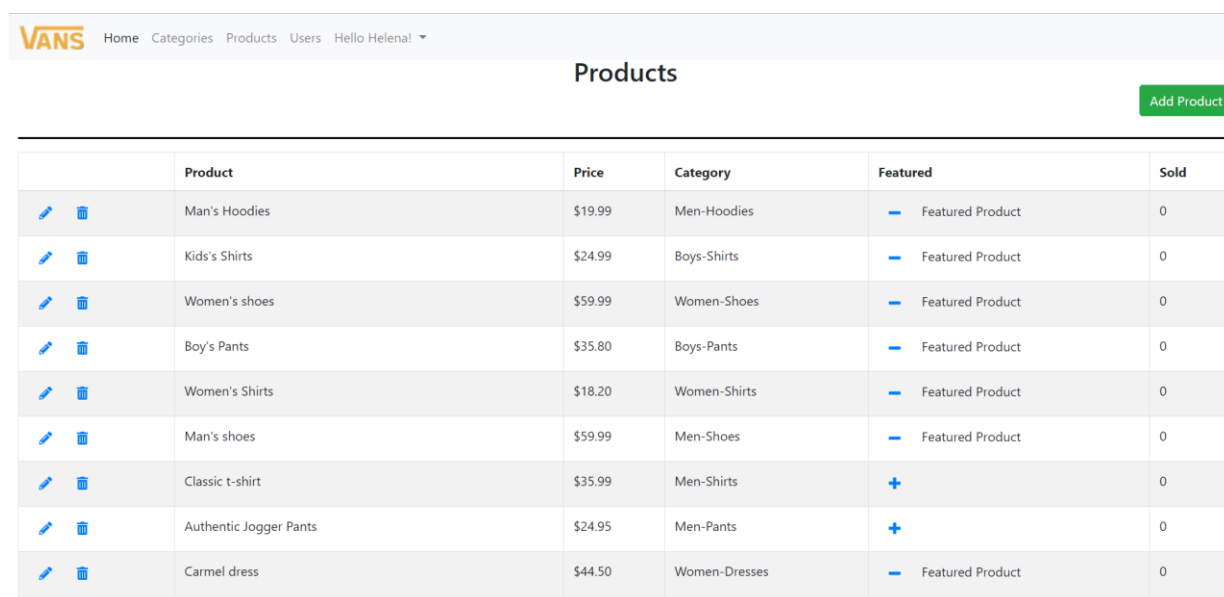
Přihlášení do sekce správy webu je implementováno za použití webového formuláře, který je umístěn ve středu stránky. Formulář obsahuje dvě vstupní textová pole s jejich popisem, tlačítko pro potvrzení a odeslání formuláře a textový řetězec v podobě odkazu: “Visit Site.“, za pomoci něhož se uživatel proklikne zpět na úvodní stranu internetového obchodu. První vstupní textové pole slouží pro zadání e-mailové adresy uživatele a druhé vstupní pole poté pro zadání hesla uživatele. Pro úspěšně přihlášení je vyplnění hodnot do obou vstupních polí povinné. Jakmile jsou obě pole vyplněna, uživatel formulář potvrdí kliknutím na odesílací tlačítko s názvem: “Login“. Systém nejprve provede kontrolu splnění podmínky, která vyžaduje vyplnění obou textových polí a také formu textového vstupu, kterou zde uživatel zadal. E-mailovou adresu je nutno napsat v korektní formě zápisu, dle stanovených zvyklostí, tzn. místní část adresy (např. hela.urbaskova) následuje znak závináče (@) a poté internetová doména. Následně systém v pozadí provede validaci dat v porovnání s interní databází, tzn. zdali uživatel zadal jméno a k němu korektní heslo, které je evidováno v databázi. Po úspěšné validaci veškerých dat a jejich forem je uživatel úspěšně přihlášen do administrátorské části webu.





















Obrázek 4.21: Přihlašovací formulář [Interní zdroj]

4.2.1.5. Správa webu – Products

Pro přidání, editaci a úpravu nabídky produktů je zde implementována sekce Products v administrativní části webu. Pomocí zeleného tlačítka umístěného v levé horní části stránky bude zákazník přesměrován na stránku obsahující formulář uzpůsobený pro přidání nového produktu. Zobrazení produktu je implementováno pomocí tabulky se seznamem veškerých produktů. Tabulka je složena ze šesti sloupců, přičemž první sloupec zleva obsahuje dvě vektorové ikony – tužku a koš, pro editaci a smazání vybraného produktu. Následující sloupec obsahuje název produktu. Dále se zde nachází cena zboží za kus, kategorie zboží a hodnota atributu Featured, který charakterizuje, zdali je produkt přítomný ve výběru produktů na úvodní stránce. Znaménko + charakterizuje možnost přidání produktu na úvodní stránky. Znaménko – je znakem toho, že se již daný produkt na úvodní stránce uchází. Poslední sloupec udává počet prodaných kusů daného zboží.

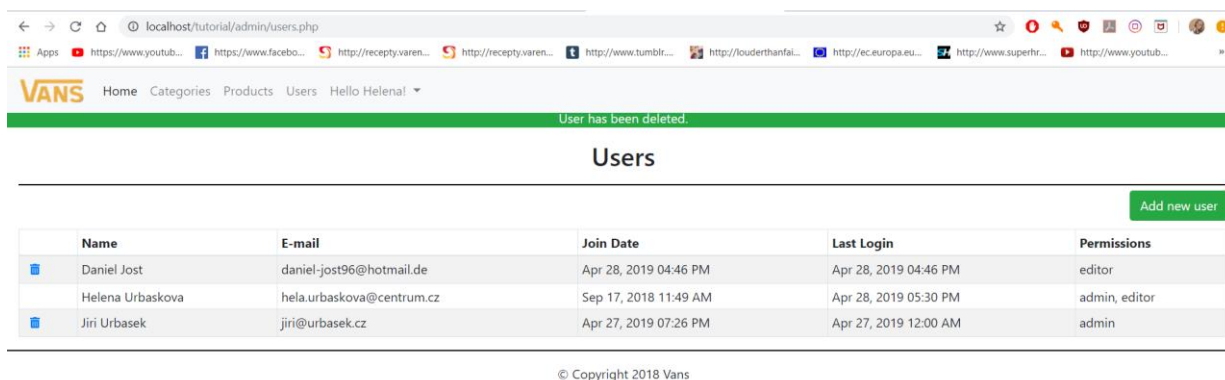


Products						Add Product
	Product	Price	Category	Featured	Sold	
 	Man's Hoodies	\$19.99	Men-Hoodies	— Featured Product	0	
 	Kids's Shirts	\$24.99	Boys-Shirts	— Featured Product	0	
 	Women's shoes	\$59.99	Women-Shoes	— Featured Product	0	
 	Boy's Pants	\$35.80	Boys-Pants	— Featured Product	0	
 	Women's Shirts	\$18.20	Women-Shirts	— Featured Product	0	
 	Man's shoes	\$59.99	Men-Shoes	— Featured Product	0	
 	Classic t-shirt	\$35.99	Men-Shirts	+	0	
 	Authentic Jogger Pants	\$24.95	Men-Pants	+	0	
 	Carmel dress	\$44.50	Women-Dresses	— Featured Product	0	

Obrázek 4.22: Produkty [Interní zdroj]

4.2.1.6. Správa webu – Uživatelé

V případě úspěšného přihlášení uživatele a za předpokladu, že má daný uživatel status „Admin“ je poté k dispozici sekce správy webu Users. V levém horním logu se nachází logo firmy, vedle nějž je umístěn navigační panel webu. Menu obsahuje seznam odkazů, prostřednictvím kterého se lze při kliknutí na daný název odkazu prokliknout na námi požadovanou stránku.



Obrázek 4.23: Uživatelé [Interní zdroj]

4.2.1.7. Změna hesla

Změnu hesla je možno provést v sekci správy webu kliknutím na odkaz v horní části navigačního panelu, který je podrobněji popsán v kapitole 4.2.1.7.2 Správce webu – Uživatelé. Po kliknutí dojde k přesměrování na níže vyobrazenou stránku. V prostřední části stránky se nachází webový formulář, který nese název: “Change password.” a v rámci něhož jsou obsaženy tři vstupní textová pole a dvě tlačítka. První textové pole slouží k zadání starého hesla uživatele, druhé pole k zadání nového hesla a třetí textové pole vyžaduje opětovné zadání nového heslo pro kontrolu správnosti zadané hodnoty v poli číslo dvě. Pod těmito vstupními poli jsou dostupná dvě tlačítka – první tmavě žluté barvy s nápisem: “Cancel” nás v případě, že bychom v průběhu vyplňování formuláře změnili názor a heslo přece jen chtěli ponechat stejné, přesměruje na úvodní stránku administrátorské části webu. Tlačítko umístěné vedle tlačítka „Cancel“, jehož popisek: “Change” nám již napovídá, že bude sloužit k potvrzení formuláře a odeslání dat na server a požadavku do databáze.

Change password

Old Password:

New Password:

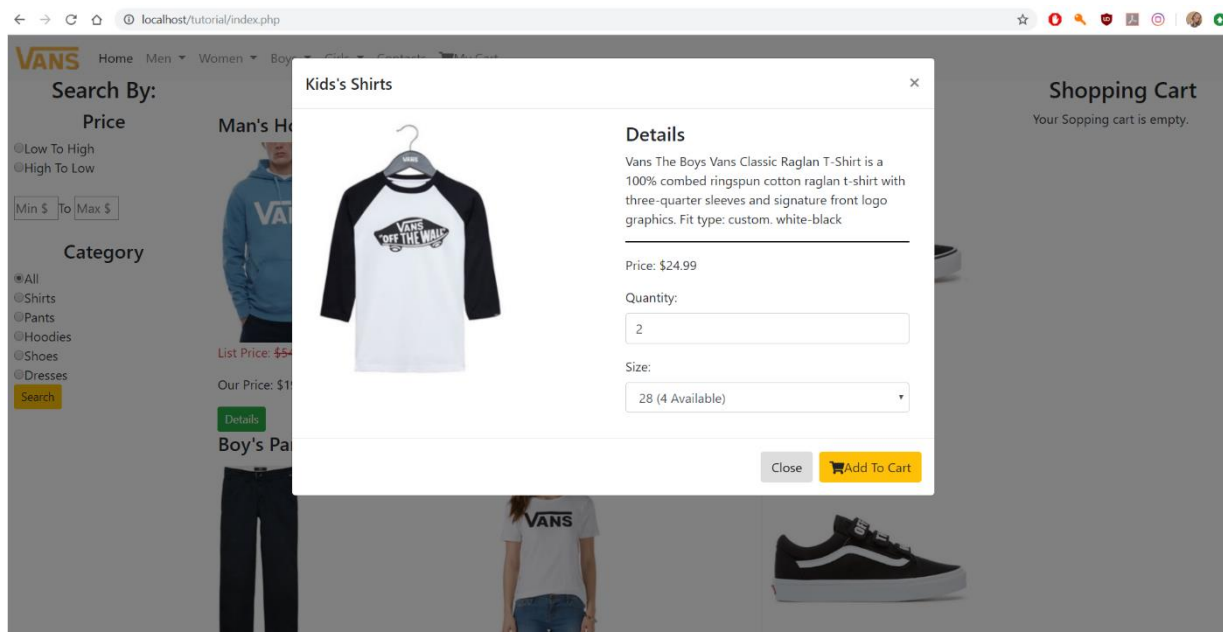
Confirm New Password:

[Visite Site](#)

Obrázek 4.24: Formulář změna hesla [Interní zdroj]

4.2.2. Detail položky

4.2.2.1. Popis



Obrázek 4.25: Detail položky [Interní zdroj]

K zobrazení podrobnějších informací o námi vybraném produktu jsem zde pomocí Bootstrap knihovny implementovala vyskakovací okno („Bootstrap Modal Plugin“) využívající HTML, CSS a programovacího jazyka JavaScript.

```
<button type="button" class="btn btn-sm btn-success"
onclick="detailsmodal( <?php echo $product['id'];?> )
">Details</button>
```

Zdrojový kód 4.1: Details modal [vlastní zdroj]

Ke spuštění modulu se v dnešní době využívá tlačítko anebo odkaz. V našem případě jsem zvolila zelené tlačítko s intuitivním nápisem „Detail“, které zákazníka navádí ke kliknutí na něj v případě, že by se rád o produktu dozvěděl více informací s ideálním výsledkem přidání daného produktu do svého košíku.

JS onclick Event neboli Událost „na kliknutí“ implementována v jazyce JavaScript se již podle názvu využívá v případě, že chceme provést kód JavaScriptu při kliknutí na element. Za pomoci parametru JS metody detailsmodal, kterou jsme si deklarovali v rámci patičky (footer.php) se nám bude obsah genericky měnit na základě ID daného produktu.

V kódu zobrazeném níže deklarujeme JS funkci, ve které nejprve vytváříme proměnnou data, která se rovná JSON objektu a jehož hodnota parametru s názvem „id“ se rovná našemu ID produktu, tzn. parametru naší detailsmodal funkce (metody).

Dalším krokem je prostřednictvím knihovny jQuery volání metody `ajax()`, která slouží k provedení tzv. asynchronního AJAX požadavku vůči serveru, pro získání dat o produktu z naší databáze. Tělo našeho JSON objektu se skládá z: url = cesta k souboru, kde chceme data obdržet; HTTP metody = post, která je využívána pro posílání dat na server za účelem vytvoření/modifikace daného zdroje (potvrzený formulář, nahrávání souborů); data = informace o našem produktu (v kódu data dle daného ID); success = V případě pozitivního scénáře požadována data vložíme na konec našeho těla a následně pomocí ID lokalizujeme náš modul, který za pomoci JS metody modal („toggle“) otevřeme.; error = V případě že nastane v průběhu požadavku chyba zobrazí se nám menší okno s nápisem „Something went wrong.“.

```

function detailsmodal(id) {
    var data = {"id":id};
    jQuery.ajax({
        url: "/tutorial/includes/detailsmodal.php",
        method : "post",
        data : data,
        success : function(data) {
            jQuery('body').append(data);
            jQuery('#details-modal').modal('toggle');
        },
        error : function() {
            alert("Something went wrong!");
        }
    });
}

```

Zdrojový kód 4.2: Details Modal [vlastní zdroj]

Vzhledem k tomu, že výběr a následné užití veškerého kódu ze souboru „detailsmodal.php“ by bylo zbytečné a značně neefektivní, použijeme zde PHP metodu `ob_start()`, která nám umožní ukládat výstup do vyrovnávací paměti (buffer) a poté jej zašle ve formě datového objektu zpět jako odpověď na AJAX požadavek. V momentě, kdy již daná data nepotřebujeme buffer opět pomocí PHP metody `ob_get_clean()` vyprázdníme.

Abychom obdrželi ta správná data na základě ID produktu, je nutno provést následující část kódu, ve které přistupujeme k poli provedené metody POST a zároveň se také ujistujeme, že vrácená hodnota ID bude ve tvaru INT namísto jakéhokoliv formátu, který nám byl předán skrze metodu POST. Za pomoci MySQL dotazu poté deklarujeme přiřazení našeho ID z databáze k proměnné `$id` a prostřednictvím provedení zbylého kódu zařídíme dynamické generování obsahu na základě Product ID.

```

require_once '../system_core/init.php';
$id = $_POST['id'];
$id = (int)$id;
$sql = "SELECT * FROM products WHERE id = '$id' ";
$result = $db->query($sql);
$product = mysqli_fetch_assoc($result);

```

Zdrojový kód 4.3: Details Modal – SQL query [vlastní zdroj]

4.2.2.2. Provedení objednávky

4.2.2.3. Popis

Celý proces provedení objednávky začíná udržením zákazníka na stránce našeho internetového obchodu. Zákazník má díky výše popsanému vyskakovacímu oknu možnost, si zobrazit více detailů o daném produktu. Pomocí tlačítka „Add to cart“ a výběru vhodné velikosti a počtu kusů zboží dojde k zavolání javascriptové funkce `add_to_cart()`. V první kroku si uložíme veškeré hodnoty zadané zákazníkem do příslušných proměnných. Poté si do proměnné `data` uložíme uvedená data formuláře, které jsme pomocí jQuery metody `serialize()` převedli do formy textového řetězce dle standardního URL kódování. Serializované hodnoty poté můžeme poslat v rámci AJAX požadavku na server v rámci dotazovacího řetězce, který je zobrazen jako součást URL adresy v podobě páru klíč-hodnota následuje znak otazníku (?), v případě předání více hodnot jsou tyto páry od sebe odděleny znakem ampersandu (&).

```
function add_to_cart(){
    $('#modal_errors').html("");
    var size = $('#size').val();
    var quantity = $('#quantity').val();
    var available = $('#available').val();
    var error = '';
    var data = $('#add_product_form').serialize();
```

Zdrojový kód 4.4: Cart [vlastní zdroj]

Následně provedeme kontrolu vstupních údajů – zdali jsme zadali veškeré potřebné údaje; zdali počet kusů zboží nepřekračuje hranici dostupných kusů na skladě. Za předpokladu, že zadaná data jsou validní, dojde k jejich předání prostřednictvím AJAX požadavku do php dokumentu s názvem `add_cart.php`. V opačném případě se nám zobrazí chybové hlášení s popisem chyby. Pro korektní nastavení cookies je nutno stránku opětovně načíst, proto je zde použita metoda `location.reload()`.

```
if(size == '' || quantity == '' || quantity == 0){
    error += '<p class="text-danger text-center">You must
choose a size and quantity!</p>';
    $('#modal_errors').html(error);
    return;
} else if(quantity > available){
```



```

    error += '<p class="text-danger text-center">There are only
'+available+' available.</p>';

    $('#modal_errors').html(error);

    return;
} else {
    $.ajax({
        url : '/tutorial/admin/parsers/add_cart.php',
        method : 'post',
        data : data,
        success : function() {
            location.reload();
        },
        error : function() {alert("Something went wrong");

```

Zdrojový kód 4.5: Cart – Add to cart [vlastní zdroj]

K uložení hodnoty ID daného nákupu použijeme cookies.

```

define('CART_COOKIE','SBiki85UCklwiqzz2');
define('CART_COOKIE_EXPIRE',time() + (86400 * 30));
$cart_id = '';
if(isset($_COOKIE[CART_COOKIE])) {
    $cart_id = sanitize($_COOKIE[CART_COOKIE]);
}

```

Zdrojový kód 4.6: Cart - Cookies [vlastní zdroj]

Prostřednictvím asociativního pole si do databáze uložíme hodnotu ID vybraného produktu, velikost a počet kusů dle objednávky.

id	items	expire_date	paid	shipped
2	[[{"id":"5","size":"Small","quantity":"3"}]]	2018-04-21 13:08:21	0	0

Obrázek 4.26: Databázový náhled na konkrétní nákup [vlastní zdroj]

Za pomoci příkazu `SELECT` vybereme daný produkt/y na základě `Product ID` a ten poté uložíme za po užití `Relace(Session)`. V případě, že uložení hodnoty proběhne v pořádku, zobrazí se uživateli zpráva obsahující „Name of the product(`title`) was added to your cart.“

```
$query = $db->query("SELECT * FROM products WHERE id =  
'{$product_id}'");  
  
$product = mysqli_fetch_assoc($query);  
  
$_SESSION['success_flash'] = $product['title']. ' was added to  
your cart.';
```

Zdrojový kód 4.7: Cart – SQL query [vlastní zdroj]

V dalším kroku ověřujeme, zdali `$cart_id`, což symbolizuje ID nákupu daného uživatele, již existuje. Tento krok se provádí z toho důvodu, abychom v případě přidání více položek do košíku v rámci jednoho nákupu pokaždé obsah košíku pouze aktualizovali (navýšili počet položek) a nevytvářeli obsah košíku pokaždé od začátku.

V případě, že cookie pro daný nákup již existuje, tak pomocí `foreach()` smyčky zkontrolujeme, zdali se již jedna z předchozích položek košíku a hodnota velikosti daného zboží shodují s nově přidanou položku. Jestliže ano pouze množství daného zboží zvýšíme o 1, jestliže ne přidáme novou položku do seznamu položek. Ošetřena je zde také možnost přidání většího množství zboží daného produktu než je dostupné na skladě. Následně pomocí příkazu `UPDATE` aktualizujeme databázi a nastavíme parametr `expire` metody `setcookie()` = 1, což symbolizuje vypršení aktuální hodnoty cookies.

V případě, že zde žádná hodnota cookie nastavena není, provede se následující algoritmus.

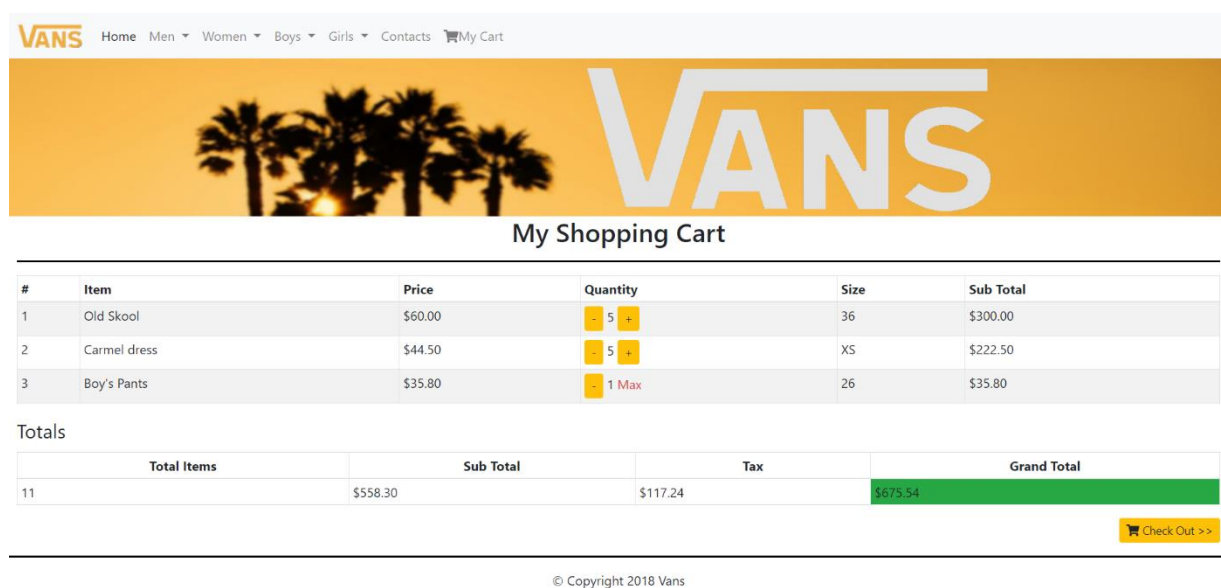
Přidáme obsahu košíku do databáze, tabulka cart a nastavení cookie v prohlížeči uživatele. Obsah našeho košíku převedeme použitím `json_encode()` metody na JSON formát dat a v databázi ho také můžeme v tomto formátu vidět. Nastavení cookies do webového prohlížeče provedeme prostřednictvím metody

```
setcookie(name,value,expire,path,domain,secure);
```

```
else{
    $items_json = json_encode($item);
    $cart_expire = date( "Y-m-d H:i:s",strtotime("+30 days")
);
    $db->query( "INSERT INTO cart (items,expire_date) VALUES
( '{$items_json}','{$cart_expire}')" );
    $cart_id = $db->insert_id;
    setcookie(CART_COOKIE,$cart_id,CART_COOKIE_EXPIRE,'/', $dom
ain,false);
}
```

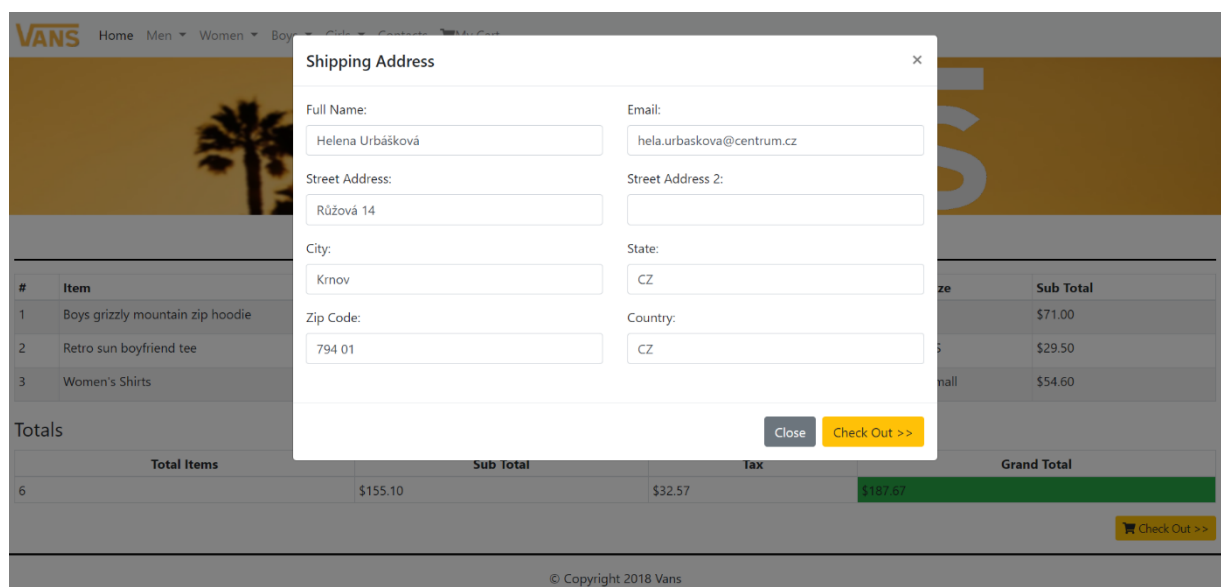
Zdrojový kód 4.8: Cart – SQL query, insert into cart [vlastní zdroj]

Položky je poté možné zobrazit kliknutím na ikonku košíku a nápisu My Cart. Zákazník je přesměrován na novou stránku. Pro každou položku je zde uveden název produktu, vybrané množství zákazníkem, které je zde také ještě možno měnit. Dále cena za ks, velikost a cena celkem za daný počet kusů zboží. Ve spodním řádku je poté uveden počet ks zboží, cena před zdaněním, výše ceny daně a cena zvýšena o daň.



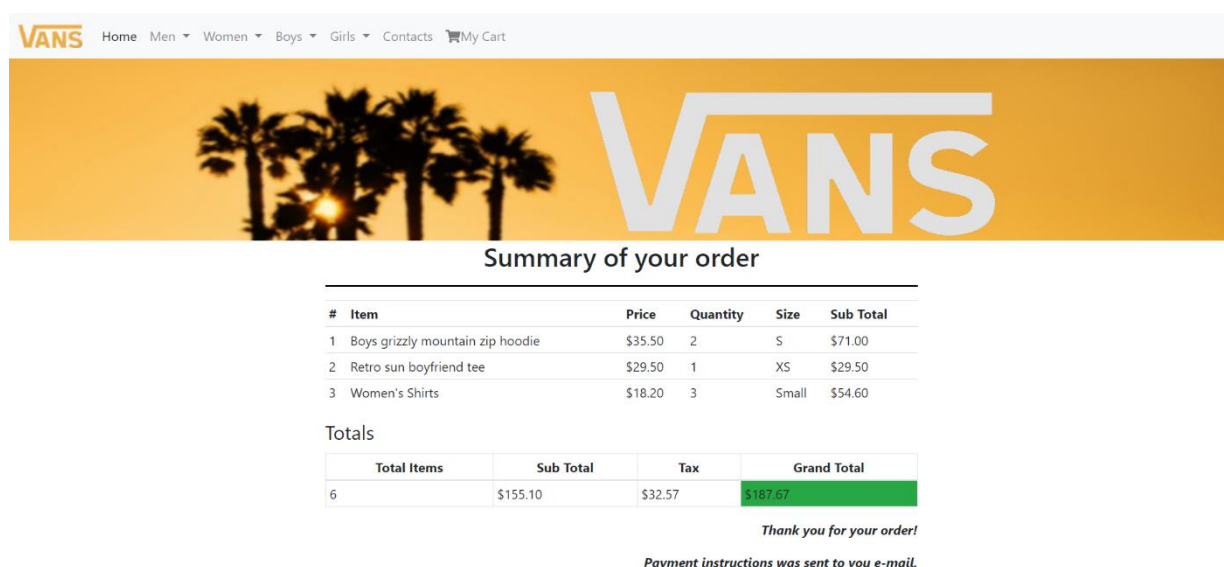
Obrázek 4.27: Náhled košíku [vlastní zdroj]

Kliknutím na tlačítko v pravém spodním rohu „Check out“ systém zobrazí nové vyskakovací okno, obsahující formulář s informacemi o adrese zákazníka. V případě, že zákazník nevyplní validní data zobrazí se chybová hláška s popisem chyby, př. Zadan nevalidní formát e-mailu apod.



Obrázek 4.28: Náhled formuláře – shipping address [vlastní zdroj]

Jestliže jsou vstupní data validní následuje zobrazení shrnutí objednávky, poděkování a zaslání informací o platebních instrukcích na e-mail zákazníka.



Obrázek 4.29: Náhled shrnutí objednávky [vlastní zdroj]

5. Závěr

První kapitola se věnuje krátkému shrnutí současné situace v podnikatelské sféře, týkající se živnostenské formy podnikání, technologickému rozvoji a jeho vlivu na zvyšující se požadavky v boji o zákazníka. Druhou část tvoří definování cílů této bakalářské práce.

Druhá kapitola „Teoretická východiska návrhu a vývoje e-shopu“ se z počátku zabývala pohledem do historie Internetu. Po tomto stručném úvodu k tématu jsme již přešli k informacím, které se týkají praktické, implementační části e-shopu. Mezi popsání implementační technologie se řadí HTML, ke kterému neodmyslitelně patří CSS, jejímž spojením vznikla knihovna Bootstrap stavebních bloků kódu velice jednoduše implementovatelná již se základními znalostmi HTML. K aplikaci dynamického přístupu k webovým stránkám jsme použili programovací jazyky PHP a JavaScript, který jsme implementovali také pomocí knihovny jQuery. Vzhledem k tomu, že by se žádný e-shop neobešel bez databázového řešení, použili jsme v praxi často užívané řešení pomocí aplikace phpMyAdmin a dotazovacího jazyka MySQL. Důležitá část dynamického přístupu zpracování obsahu je jeho komunikace se serverem, s tímto tématem souvisí popis technologie AJAX v kombinaci s datovým formátem JSON.

Třetí kapitola „Analýza současného stavu“ obsahuje stručný popis společnosti. Dále soubor poznatků a informací nezbytných pro následné definování požadavků zadavatelky. Analýzu požadavků a jejich následné rozdělení na požadavky funkční a nefunkční a také analýzu konkurenčního prostředí, čímž byla splněna první kategorie cílů.

Čtvrtá kapitola „Návrh a implementace e-shopu“ se věnuje stěžejní části této práce, kterou je návrh, vývoj a implementace daného řešení. V rámci podkapitoly Návrh jsme nejprve definovali vhodnou metodiku pro návrh a vývoj internetového obchodu. Následně byly specifikovány a modelovány konkrétní případy užití s využitím UML diagramů. Návrh databázové struktury a uživatelského rozhraní se nachází v podkapitolách 4.1.3 a 4.1.4. Splnění cílů kategorie Návrh jsme zakončili specifikací aplikovaného softwaru. Druhá část této kapitoly byla věnována dosažení cílů definovaných v rámci kategorie Implementace, kde je popsána implementace uživatelského rozhraní a jsou zde obsaženy ukázky implementace vybraných funkcionalit.

Tímto projektem naše spolupráce se slečnou Rabiečnou rozhodně nekončí, jelikož je zde stále prostor pro zlepšení funkcionalit e-shopu. V budoucnu bychom rádi přidali např. možnost filtrování produktů dle různých kritérií, možnost filtrování objednávek a další funkcionality.

Seznam použité literatury

Knižní zdroje

1. SCHAFER, Steven M. *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek]: 4. vydání*. Praha: Grada, 2009. Průvodce (Grada). ISBN 9788024728506.
2. STAUFFER, Todd. *Tvorba webových stránek pro úplné začátečníky*. Praha: SoftPress, 2003. ISBN 80-86497-38-0.
3. GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. *HTML5 a CSS3 pro webové designéry*. Brno: Zoner Press, 2011. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.
4. STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. *Naučte se SQL za 28 dní: [stačí hodina denně]*. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1
5. PROCHÁZKA, David. *CSS A XHTML: tvorba dokonalých WWW stránek krok za krokem*. 2. vyd. Praha: Grada, 2011. ISBN 978-80-247-3897-0.
6. ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
7. RUMBAUGH, James, Ivar JACOBSON a Grady BOOCH. *The unified modeling language reference manual*. 2nd ed. Boston: Addison-Wesley, c2005. ISBN 0-321-24562-8.
8. MIKULÁŠKOVÁ, Petra a Mirek SEDLÁK. *Jak vytvořit úspěšný a výdělečný internetový obchod*. Brno: Computer Press, 2015. ISBN 978-80-251-4383-4.

Internetové + Ostatní zdroje

9. Peterka, Jiří. Na počátku byl ARPANET. eArchiv.cz. [online] 1. 4 1995. [cit. 2009-04-12] <<http://www.earchiv.cz/a95/a504c502.php3>>.
10. KÖNIG, Tomáš. *Návrh a implementace internetového obchodu*. Ostrava, 2017. Bakalářská práce. Vysoká škola báňská – Technická univerzita Ostrava, Fakulta ekonomická, Katedra aplikované informatiky.

11. Navrhování responzivních webů pomocí nástroje Bootstrap. 301 Moved Permanently [online]. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/bootstrap.html>
12. Bootstrap – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
13. phpMyAdmin - Wikipedia. [online]. Dostupné z: <https://en.wikipedia.org/wiki/PhpMyAdmin>
14. XAMPP – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/XAMPP>
15. JSON. JSON [online]. Dostupné z: <https://www.json.org/json-cz.html>
16. VOTAVA, Jiří. Realizace e-shopu pro malou firmu. Ostrava 2012. Diplomová práce. Vysoká škola báňská – Technická univerzita Ostrava, Fakulta ekonomická, Katedra aplikované informatiky.
17. MVŠO - Moravská vysoká škola Olomouc [online]. Copyright © [cit. 04.05.2019]. Dostupné z: <https://mvso.cz/wp-content/uploads/2018/02/Metodiky-v%C3%BDvoje-software-studijn%C3%AD-text.pdf>
18. All You Need to Know About UML Diagrams: Types and 5+ Examples. Tallyfy: Workflow Software, Business Process Management Software [online]. Copyright © 2015 [cit. 05.05.2019]. Dostupné z: <https://tallyfy.com/uml-diagram/>

Seznam zkratek

API	Application Programming Interface
ARPA	Advanced Research Project Agency
CERN	Evropské centrum pro jaderný výzkum
CESNET	Czech Educational and Scientific NETwork
CSS	Cascading Style Sheets
CROS	Cross-origin resource sharing
ČVUT	České vysoké učení technické
DOM	Document Object Model
DOCTYPE	Document Type
GPL	General Public License
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ID	Identifier
IP	Internet Protocol
LAMP	balík obsahující trojici programů Apache+MySQL+PHP pro operační systém Linux
MySQL	My Structured Query Language
NSFNET	National Science Foundation NETwork
PHP	Hypertext Preprocessor
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
W3C	World Wide Web Consortium
WWW	World Wide Web
XAMPP	softwarový balík složený z Apache+MariaDB+Perl X=Multiplatformní
XHTML	Extensible hypertext markup language
XML	eXtensible Markup Language

Seznam příloh

Příloha č.1: Scénáře případů užití

UC1 – Přihlášení

Tabulka 1: Scénář případu užití – UC1, zdroj: vlastní

Název:	Přihlášení
ID:	UC1
Aktéři:	Správce internetového obchodu nebo editor
Vstupní podmínky:	Potřeba administrátora či editora provést akci vyžadující přihlášení do systému
Hlavní tok:	Případ užití spouští Správce internetového obchodu nebo editor (dále uživatel) 1. Uživatel otevře webovou stránku pro přihlášení 2. Systém zobrazí přihlašovací formulář 3. Uživatel vyplní objednávkový formulář 4. Systém provede validaci zadaných hodnot 6. KDYŽ zadané hodnoty splňují předepsaný formát dle formuláře 6.1. Systém provede přihlášení 6.2.2 Editor – Zobrazení úvodní stránky dle editorských práv 6.2.3 Admin – Zobrazení úvodní stránky dle administrátorských práv 7. NEBO 7.1. Systém zobrazí chybové hlášení 7.2. PŘESUN NA krok č. 3
Error msg:	1. Systém kontroluje validaci vložených dat 2. KDYŽ není zadán validní e-mail 2.1. Systém odpoví chybovým hlášením 3. KDYŽ není zadáno validní heslo 3.1. Systém odpoví chybovým hlášením 4. KDYŽ nejsou zadány veškeré povinné údaje 4.1. Systém odpoví chybovým hlášením 5. KDYŽ jsou data validní 5.1. Úspěšně přidání uživatele
Následné podmínky:	Úspěšně přihlášení do administrace
Cíl:	Umožněním vstupu do admin.části webu dostupná možnost editace obsahu webu

UC5 – Editace produktu

Tabulka 2: Scénář případu užití – UC5, zdroj: vlastní

Název:	Editace produktu
ID:	UC5
Aktéři:	Správce internetového obchodu nebo editor
Vstupní podmínky:	Potřeba administrátora či editora editaci daného produktu a úspěšné přihlášení do systému
Hlavní tok:	<p>Případ užití spouští Správce internetového obchodu nebo editor (dále uživatel)</p> <ol style="list-style-type: none"> 1. Uživatel otevře stránku Products 2. Systém zobrazí seznam produktů, ikony pro editaci a smazání 3. Uživatel kliknutím na ikonu tužky vybere možnost editace produktu 4. Systém zobrazí vyplněný formulář s daty o daném produktu 5. Uživatel pozmění požadované pole informací 6. Systém provede validaci zadaných hodnot 6. KDYŽ zadané hodnoty splňují předepsaný formát dle formuláře <ol style="list-style-type: none"> 6.1. Systém provede aktualizaci databáze 7. NEBO <ol style="list-style-type: none"> 7.1. Systém zobrazí chybové hlášení 7.2. PŘESUN NA krok č. 5
Error msg:	<ol style="list-style-type: none"> 1. Systém kontroluje validaci vložených dat 2. KDYŽ nejsou zadána povinná pole 3. KDYŽ jsou data validní <ol style="list-style-type: none"> 3.1. Úspěšná aktualizace dat v databázi
Následné podmínky:	Úspěšně ověření změny dat na webu
Cíl:	Provedena požadovaná změna dat u daného produktu

UC9 – Přidání uživatele

Tabulka 3: Scénář případu užití – UC9, zdroj: vlastní

Název:	Přidání uživatele
ID:	UC9
Aktéři:	Správce internetového obchodu
Vstupní podmínky:	Potřeba přidání nového uživatele
Hlavní tok:	<p>Správce internetového obchodu (dale Admin)</p> <ol style="list-style-type: none"> 1. Admin klikne na „Create new user“ v administrátorské části - Users 2. Systém provede aktivitu “Enter required data” 3. Admin Zadá požadovaná data 4. Admin Potvrdí data 5. Systém provede validaci zadaných hodnot 6. KDYŽ zadané hodnoty splňují předepsaný formát dle formuláře

	6.1. Systém provede aktivitu "Add new user" 7. NEBO 7.1. Systém zobrazí chybové hlášení 7.2. Admin provede aktivitu "Refilling all informations" 7.3. PŘESUN NA krok č. 4
Error msg:	1. Systém kontroluje validaci vložených dat 2. KDYŽ není zadán validní e-mail 2.1. Systém odpoví chybovým hlášením 3. KDYŽ není zadáno validní heslo 3.1. Systém odpoví chybovým hlášením 4. KDYŽ se heslo a potvrzovací heslo neshodují 4.1. Systém odpoví chybovým hlášením 5. KDYŽ nejsou zadány veškeré povinné údaje 5.1. Systém odpoví chybovým hlášením 6. KDYŽ jsou data validní 6.1. Úspěšně přidání uživatele
Následné podmínky:	Přidání nového uživatele
Cíl:	Přidáním nového uživatele umožnit přístup dalšímu zaměstnanci

UC13 – Přidání zboží do košíku

Tabulka 4: Scénář případu užití – UC13, zdroj: vlastní

Název:	Přidání zboží do košíku
ID:	UC13
Aktéři:	Zákazník
Vstupní podmínky:	Rozhodnutí o nákupu daného produktu
Hlavní tok:	Zákazník 1. Zákazník si v rámci kategorie vybere daný produkt 2. Kliknutí na tlačítko "Details" 3. Zobrazení vyskakovacího okna s detailním popisem zboží 4. Výběr velikosti a počtu kusů zboží 4.1. Výběr většího počtu kusů zboží než je momentálně k dispozici 4.2. Systém zobrazí chybové hlášení 5. NEBO 5.1. Zákazník vybere adekvátní zboží vzhledem k dostupnosti zboží 5.2. Přidání položky do košíku pomocí tlačítka "Add to cart" 6. Systém přidá danou položku do košíku
Následné podmínky:	Přidání zboží do košíku
Cíl:	Přítomnost položky a validní počet kusů a velikost v košíku

UC14 – Potvrzení objednávky

Tabulka 5: Scénář případu užití – UC14, zdroj: vlastní

Název:	Potvrzení objednávky
ID:	UC14
Akteři:	Zákazník
Vstupní podmínky:	Úspěšné provedení UC11 a UC13, potvrzení rozhodnutí o nákupu
Hlavní tok:	<p>Zákazník</p> <ol style="list-style-type: none"> 1. Zákazník úspěšně provede UC11 a také UC13 2. Kliknutí na tlačítko „Check out” 3. Systém zobrazí vyskakovací okno s formulářem obsahujícím doručovací údaje 4. Zákazník nevyplní daná pole dle požadovaného formátu <ol style="list-style-type: none"> 4.1. Systém zobrazí chybové hlášení 5. NEBO <ol style="list-style-type: none"> 5.1. Zákazník vyplní veškeré údaje ve správném formátu 6. Zákazník klikne na tlačítko “Submit an order” 7. Systém přesměruje zákazníka na stránku “ThankYou.php” a zobrazí zákazníkovi přehled jeho objednávky
Následné podmínky:	Platná objednávka
Cíl:	Potvrzení objednávky zákazníkem

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- беру на вѣдомі, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 10.5.2019



jméno a příjmení studenta